Self-Attention Based Network for Punctuation Restoration

Feng Wang*†, Wei Chen*, Zhen Yang*and Bo Xu*
*Institute of Automation, Chinese Academy of Sciences
No.95 Zhongguancun East Road
{feng.wang, wei.chen.media, yangzhen2014, xubo}@ia.ac.cn
† University of Chinese Academy of Sciences, Beijing, China

***Abstract*—Inserting proper punctuation into Automatic Speech Recognizer(ASR) transcription is a challenging and promising task in real-time Spoken Language Translation(SLT). Traditional methods built on the sequence labelling framework are weak in handling the joint punctuation. To tackle this problem, we propose a novel self-attention based network, which can solve the aforementioned problem very well. In this work, a light-weight neural net is proposed to extract the hidden features based solely on self-attention without any Recurrent Neural Nets(RNN) and Convolutional Neural Nets(CNN). We conduct extensive experiments on complex punctuation tasks. The experimental results show that the proposed model achieves significant improvements on joint punctuation task while being superior to traditional methods on simple punctuation task as well.**

I. Introduction

Automatic Speech Recognizer (ASR) typically produces a sequence of words without punctuation, which is difficult to read for humans. The punctuation often plays a crucial role in understanding the semantics of these sentence and it's vital for the downstream NLP tasks, such as question answering, machine translation, sentiment analysis, and information extraction. Therefore, it is challenging and promising to investigate the problem of punctuation restoration.

Recently, some previous research on punctuation restoration has been proposed, which can be mainly classified into three branches: *pure lexical features based*, *acoustic features based* and *hybrid features based*. The pure lexical features based approaches utilize the textual features only and without relying on the audio features. The acoustic features based approaches rely entirely on the audio or acoustic features, such as pitch, intensity and pause duration. And the hybrid features based approaches combine the textural and audio features, which achieve better performance. For example, the approaches for combing textual and prosodic features into decision tree to predict like Conditional Random Fields (CRFs) or adaptive algorithm [1], [2].

Naturally, a pure lexical features based model is less powerful than a hybrid features based model[3], [4]. However, the training data for the hybrid features based model is scarce and uneasily reachable because it must be the standardized ASR transcripts. On the contrary, the pure lexical models can take any kind of textual material as training data and can be freely used in late fusion with acoustic features. Therefore, researchers believe that a good lexical model is widely applicable. This paper also takes the lexical features only and aims to build stronger lexical model.

In the lexical features based branch, various methods have been proposed, like n-gram models [5], transition-based dependency parsing[6], [7], CRFs [8], [9] and deep neural network [10], [11], [12]. Some of the systems use the encoder-decoder framework with attention mechanism, which has been widely used in many sequence-to-sequence translation tasks [13]. These approaches achieve some success by using the textual features. However, these approaches have a fatal weakness in handling the joint punctuation, such as @,:@",''@:"@ and so on [1], which have more complex semantics and structures than single punctuation. Moreover, for different approaches mentioned above, the researchers switch between the Recurrent Neural Network(RNN) and Convolutional Neural Network(CNN). And the RNN/CNN shows less flexibility than the self-attention layer which has been widely used in neural machine translation [14], [15] and language understanding [16].

Motivated by the analysis above, we propose a RNN/CNN-free network for restoring the joint punctuation, which is built only on the self-attention layer. In the proposed model, we apply the self-attention layer to extract the hidden features of the input sentence. And two softmax layers, namely the label softmax and word softmax, are applied to perform classifications. The label softmax is used to calculate the probability of the punctuation label, and the word softmax is applied to computing the probability of the word.

In summary, this paper has two main contributions:

- We propose a self-attention based model for the punctuation restoration task. Without relying on the RNN/CNN, the proposed model has more flexibility in sequence length and allows for more parallelization, which makes great significance.
- We conduct extensive experiments on the punctuation restoration tasks. Experimental results show that the proposed model achieves significant improvements on

[1]Wei Chen is the corresponding author of this paper

[1]the @ is only used to separate the punctuation tagging from the punctuation in text

joint punctuation task while being superior to traditional methods on simple punctuation task as well.

The rest of this paper is organized as follows. Section 2 describes related work and the background. In Section 3, we propose our approach for punctuation restoration. Experiments and results are described in Section 4. We conclude in Section 5.

II. Related Work

With the development of neural network, it shows great capability in NLP tasks, such as word segmentation, sentimental analysis, neural machine translation and so on. Nowadays, the neural network also achieves great success in the field of punctuation restoration.

In punctuation restoration, most of the models follow the architecture of word segmentation, which views the process of restoring the punctuation as the sequence labelling problem. The model architectures vary from the CNN to RNN. In the sequence labelling framework, this paper [17] presented a Long Short-Term Memory(LSTM) RNN model for restoring periods and commas in speech transcripts. Some other combine prosodic and lexical classifiers for punctuation detection in ASR system [18], [19], [20], where prosodic cues such as breaks, speech rate, pitch intonation that influence placing of punctuation marks on speech transcripts have been used. These approaches rely entirely on prosodic or audio based features, where the features they designed are much complicated.

With the emerging of the classic attention mechanism and its variants, Neural Machine Translation [21] has achieved better translation performance than the traditional Statistical Machine Translation (SMT) and gained adoption in many large-scale settings[22], [23]. Since the NMT system shows its effectiveness in sequence mapping problem, it is natural for researchers to investigate how to apply the NMT system to the problem of punctuation restoration. Several pioneers begin to take the punctuation restoration as the translation task and their models follow the architecture of NMT system. [24] built a translation system to translate from unpunctuated to punctuated text instead of a language model based punctuation prediction method, which have not practically used translation framework to predict the label sequence task. [25] have translated from lower-cased, nonpunctuated language into true-cased, punctuated language, which the output sequence is then replaced into a sequence of uppercased/lowercased words and punctuation marks. Therefore, this work is still considered as label sequence task to solve the joint punctuation. And their performance on the restoration of joint punctuation is not well.

This work also takes the punctuation restoration as the translation task and we follow the newly emerged NMT model, i.e., Transformer, which achieves state-of-the-art results on both WMT2014 English-German and WMT2014 English-French translation tasks. Different from the traditional NMT models which are usually built on the RNN or CNN, the Transformer only relies on the self-attention layers, which shows more flexibility in sequence lengths and more parallelism. We extend the Transformer and propose a novel objective for the punctuation restoration task. The proposed model achieves great success in handling the joint punctuation problem.

III. The Approach

Attention mechanism has been widely studied in the NLP community. The original Transformer framework, which achieves state-of-the-art translation performance in neural machine translation, is a novel architecture based only on attention mechanism. In this section, we will give more details about the design of our proposed sequence transduction model, namely the self-attention based network for punctuation restoring, which is referred to as ***SAPR***. Figure 1 is the graphical illustration of the proposed model, which is skillfully modified on the Transformer model. The architecture of the proposed model is described detailedly in the subsections.

A. The Input and Output

Since we consider the problem of punctuation restoration as a translation task, we prepare our input and output as the training examples of the translation. In translation, the training example consists of the source-side and target-side sequence. We conduct a translation from the non-punctuation sequence to the punctuation-restored sequence. For the proposed model, the source-side input is the non-punctuation sequence, the target-side has two output sequence, i.e., the punctuation-restored sequence and the label sequence. Considering the sentence from the monolingual data:

"Hello, can I help you?"

We transform it to the training example for our model. The source-side input is:

"Hello can I help you"

The target-side punctuation-restored sequence is represented as:

"Hello, can I help you?"

And the target-side label sequence can be represented as:

"O , O O O O O ?"

Where the 'O' indicates that the label this time step the model generates is a simple word.

The consideration behind is that either the punctuation-restored sequence or the label sequence is incapable of handling the problem of punctuation restoration independently. For the punctuation-restored sequence, it guides the model to treat each word and punctuation equally. However, for the punctuation restoration task, the model just needs to distinguish the punctuation from non-punctuation. Too careful classification will detriment the classification performance. For the label sequence, it guides the model to output the either punctuation or non-punctuation. Since the non-punctuation label 'O' dominates in the training example, which will cause overfitting during training. Therefore, The fusion of the punctuation-restored sequence and the label sequence is helpful to improve the performance.

B. The Model Architecture

The proposed model follows the architecture of Transformer, which consists of the encoder and decoder. For the detail about the Transformer, we refer the reader to [14]. We mainly describe the difference between the proposed model and the Transformer.

For the input sequence to model, we firstly transform them into the embedding sequence by looking for each token in the embedding table. We have three embedding tables with each for the source-side sequence, target-side word sequence and target-side label sequence respectively. We also add "positional encoding" to the input and output embeddings. The great benefit of positional encoding is that the model would easily learn to attend by relative positions, which add the fixed offset to k so that PE_{pos+k} can be represented as a linear function of PE_{pos}

$$PE_{(pos,2i)} = sin(pos/1000^{2i/d_{model}}) \tag{1}$$

$$PE_{(pos,2i+1)} = cos(pos/1000^{2i/d_{model}}) \tag{2}$$

where i is the dimension and pos is the position.

After the embedding layer, the multi-head attention is utilized to encode the embeddings. A multi-head attention mechanism is consist of scaled dot-product attention. We compute the output as :

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3}$$

where d_k is the dimension of the key, Q is a set of queries. The keys and values are denoted by matrices K and V. The input Q, K and V would be split N different representations and compute scaled dot-product attention for each head. The result would be concatenated and project the concatenation with a feed-forward layer. These description would be expressed as follow :

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{4}$$

$$MultiHead(Q,K,V) = Concat_i(head_1, ..., head_i) \tag{5}$$

where $W_i^Q \in \mathbb{R}_{d_{model}} \times d_k$, $W_i^K \in \mathbb{R}_{d_{model}} \times d_k$ and $W_i^V \in \mathbb{R}_{d_{model}} \times d_v$ are parameter learned in training.

In addition to attention sub-layers, the feed-forward network is fully connected each layer in encoder and decoder. This consists two linear transformations with Relu activation.

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{6}$$

Where the W_1,W_2,b_1,b_2 are the trainable weights.

The most significant difference between the proposed model and the Transformer is that the proposed model has two softmax layers after the decoder. One is the word softmax and the other is the label softmax. The word softmax outputs the probability of the target-side word sequence and the label softmax gives out the probability of the label sequence.

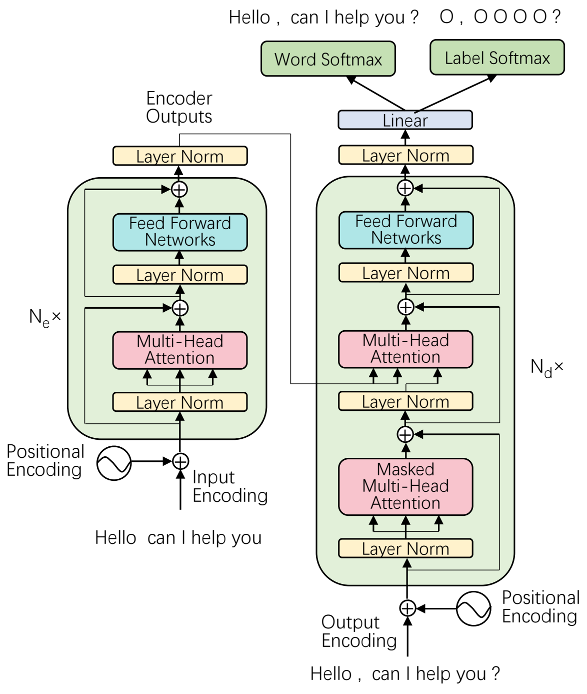

Fig. 1: The architecture of the proposed model

C. Objective Function

Given N such annotated source word sequences, label sequences and target word sequences. Our goal is to maximize the probability of observing the target word sequence y_n and minimize the label classification error rate of the label softmax, with regard to the model parameter θ. Different from the traditional End-to-End model in punctuation restoration which only maximizes the probability of observing the target label sequence, the word error rate of the word softmax is also leveraged in the proposed model. From the multi-goal learning perspective, two different goals are used to dictate the parameters of the proposed model to converge to a better region. Formally, the objective function used in the joint punctuation model can be represented as:

$$\underset{\theta}{\text{argmax}} \frac{1}{N} \sum_{n=1}^{N} (log_{p_\theta}(y_n|x_n) + \alpha * log_{p'_\theta}(z_n|x_n)) \tag{7}$$

Apart from the cross entropy loss used in the label neural network model, the error rate of the word neural network is also considered to train the proposed model.

$$J = J_m + \gamma * J_s \tag{8}$$

here J_m refers to the loss of the self-attention based neural network, J_s denotes the loss of the word and γ is the hyper-parameter which can be set by the user beforehand. We use the γ to balance the loss between the word softmax and label softmax.

Algorithm 1 Constrained decoder of the self-attention based network

```
Input: X
WordOutput: Y
LabelOutput: L
1: $step = 0$
2: $index = 0$
3: while $index < len(X)$ or $L_i \neq eos$ do
4:     for $i = 0$ to $step$ do
5:         if $Li ==' O'$ then
6:             $Yi = Xi$
7:             $index = index + 1$
8:         else
9:             $Yi = Li$
10:        end if
11:    end for
12:    $transformDecode(X, Y, L)$
13:    $step = step + 1$
14: end while
```

D. Constrained Decoder Algorithm

We design a constrained decoder algorithm to adapt the proposed self-attention based network. In this algorithm, we need to judge the class of the label softmax to determine the generation of the model. Specifically, if the label softmax outputs 'O', we copy the word from the source side and feed it into the model for the next time step. And if the output of the label softmax is not 'O', we feed the generated punctuation into the model for the decoding in the next time step. The decoding process repeats until the "eos", i.e., the end of sentence token, is emerged and all the source words are copied. The constrained decoder algorithm could be effectively avoid the inconsistency between the source word and the target word in a transduction model.

IV. Experiments

A. Datasets and Tasks

To evaluate the effectiveness of our proposed model on monolingual English punctuation restoration tasks. we implement two experiments on two different datasets. The first experiment is designed for the single punctuation task, and the second experiment is implemented to handle the joint punctuation task. Most previously published models are only evaluated on the former one.

IWSLT The dataset comes from the IWSLT TED talks, which is originally used to evaluate ASR or SLT output. So we choose the IWSLT2012[2] for the training sets and development sets. Some previous work have been done in this datasets. We use the same training, development and test set to train and test our models. We make sure that there is no overlapping between datasets. The training and development sets consist of 2.1M and 296K word respectively. More detailed description of datasets can be found in [10].

[2]http://hltc.cs.ust.hk/iwslt/index.php/evaluation-campaign/ted-task.html

AIChallenger In the second experiment, we take a large scale corpus from AI-Challenger translation task[3], which consists of 10 million parallel English-Chinese sentence pairs. The dataset is mainly from English learning websites and movie subtitles. The domain of data is spoken-language, which makes the punctuation restoration task more challenging due to the high noise and less standard of oral texts. We use the English sentences of the dataset as the training corpus for punctuation restoration and randomly extract 2000 sentences for development set and test set respectively.

All of the training sets are tokenized by the tokenizer script from the Moses decoder[4]. To speed up the training process, we clean the training data by removing the sentence pairs whose source sentence contains more than 100 words.

B. Baseline Models

We compare our model with several previously published models which achieve good performance on punctuation restoration task.

CNN Based Model For the CNN based model, we utilize the in-house framework which can be found at [5]. To give each character a label in sentence, CNN model takes the input as character sequences. We explore a CRF layer to calculate scores of label sequence and a succession of convolutional layers over embeddings to extract contextual information. Formally, considering the input character sequence x , the model projects each character into a continuous d-dimensional character embedding x using a character lookup table. In the convolutional layer, we define the number of input channels as N, the number of output channels as M, the length of input as L and kernel width is k.

$$f(x) = (x * w + b) \otimes \sigma(x * v + c) \tag{9}$$

where $W \in \mathbb{R}^{k \times N \times M}$, $b \in \mathbb{R}^M$, $V \in \mathbb{R}^{k \times N \times M}$, $c \in \mathbb{R}^M$ are parameters. And $*$ denotes row convolution operation. The input would be augmenting with paddings $f(x) \in \mathbb{R}^{L \times M}$. Then a max pooling operation is applied to extract the maximum value as the final feature of each convolutional layer's filter. The five deep convolutional layers are stacked to capture long distance information. The last layer is a linear transformation to the output of the layer to unnormalized label scores.

RNN Based Model We use a bidirectional recurrent neural network with attention mechanism [13] for the RNN based model, which has achieved the state-of-the-art on IWSLT dataset. In this model, the pre-trained word vectors(T-BRNN-Pre) are utilized to initialize the word embeddings.

C. Training Details

In our implementation, we train the proposed model on a cluster with 4 Tesla K80 GPUs and it takes about 12 hours to train the model for a total of 50000 steps. Our methods are implemented with TensorFlow [6][26]. Training ends when

[3]https://challenger.ai/competition/interpretation
[4]http://www.statmt.org/moses/
[5]https://github.com/chqiwang/convseg
[6]https://www.tensorflow.org/

Model	Period			Comma			Question			Overall		
	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1
T-LSLM	60.2	53.4	56.6	49.6	41.4	45.1	57.1	43.5	49.4	55.0	47.2	50.8
T-BRNN	72.3	71.5	71.9	64.4	45.2	53.1	67.5	58.7	62.8	68.9	58.1	63.1
T-BRNN-pre	73.3	72.5	72.9	65.5	47.1	54.8	70.7	62.8	66.7	70.0	59.7	64.4
Word CNN	75.8	95.5	84.6	56.1	60.4	58.2	40.8	55.7	47.1	65.8	77.7	71.3
SAPR	96.7	97.3	96.8	57.2	50.8	55.9	70.6	69.2	70.3	78.2	74.4	77.4

TABLE I: Results of the Single Punctuation task on IWSLT dataset

the F score on the validation set stops improving. The final models are chosen by their performance on the development set.

Input Preparation Essentially, all the baseline models treat the punctuation restoration task as classification problem by deciding whether each word in the sequence should be followed by a punctuation mark or not. In the single punctuation task, we only consider three most commonly used punctuations as most previous work does, which are period, comma and question. Each type of punctuation is treated as a class label, and label 'O' is used to indicate no punctuation mark followed. As a result, we get four classes in total. This experiment is conducted on the IWSLT dataset. In the joint punctuation task, we introduce more punctuation types, such as exclamation, semicolon, colon, quotes, and various combinations of two to four punctuations. Therefore, the models are required to be able to predict joint punctuations. As for baseline models, different combinations of punctuations are treated as different classes. Therefore, we get 41 kinds of class labels totally. This experiment is conducted on the AIChallenger dataset. Other punctuation marks in the dataset are just ignored.

BPE Byte Pair Encoding [27] is a simple data compression technique, which is highly efficient to reduce the OOV quantity by iteratively replacing the most frequent pair of bytes in a sequence with a single unused byte. In our experiments, we set source number operations to 20000 and the vocabulary size to 30K.

Optimization AdaDelta algorithm [28] is applied to optimize our models. After each updating, we clip the norm of the gradient within the block of [-1,1]. In the proposed model, the word embedding of each word in the vocabulary is regarded as part of the model's parameters, which is initialized by Gaussian distribution or uniform distribution, same as other parameters of the model.

Hyper-parameter The configuration of the hyper-parameter is critical to the performance of the proposed self-attention based model, since it directly affects the generalization and regression of the model. To balance the whole information from the word sequence and the label sequence, we introduce γ parameter. The hyper-parameter γ in equation (8) is designed to balance the loss from the label sequence and the word sequence. In order to highlight the word sequence loss, we set the γ value to 0.15 to weaken the influence of label segmentation.

D. Results on Single Punctuation Task

Table I shows the experimental results of our model and baseline models on the single punctuation task. It is clear that our novel proposed ***SAPR*** model outperforms the best bidirectional recurrent neural network(see line of T-BRNN-pre). To show the ability of our proposed ***SAPR*** model, we also compare the proposed model with the CNN based model. From table I, we find that the performance of the CNN based model is quite well, which is also superior to the best model based on bidirectional recurrent neural network. On overall performance, the CNN based model can lead to + 6.9 F value improvement than the bidirectional RNN. Moreover, the proposed ***SAPR*** model still outperforms the CNN based model. The comparison between our designed model and T-BRNN-pre model indicates that our model achieves remarkable improvements(F value increases by 11% on overall) over the state-of-the-art performance on IWSLT dataset, especially for the period punctuation. Results on other punctuation are comparable. In general, we find that the CNN model and the proposed ***SAPR*** model often achieve comparable performance and they both outperform the RNN based models. Therefore, in the next section, we would use the CNN model as the baseline for the joint punctuation experiment.

Model	CNN model			SAPR		
	Pr.	Re.	F1.	Pr.	Re.	F1.
Period	58.9	59.1	58.9	88.3	84.7	87.5
Comma	57.9	59.2	58.2	87.8	91.1	88.5
Question	39.6	40.0	39.7	90.2	86.7	89.5
Exclamation	35.5	14.8	27.0	53.8	13.0	33.0
Semicolon	33.3	37.5	34.1	80.8	63.9	76.8
Colon	–	–	–	94.6	77.8	90.3
Quotes	–	–	–	77.7	58.3	72.3
Joint Punctuation	18.4	20.1	18.8	–	–	–
Overall	55.9	55.4	55.8	87.8	85.8	87.4

TABLE II: Results of the Joint Punctuation task on AIChallenger dataset

E. Results on Joint Punctuation Task

In this experiment, we test the effectiveness of the proposed model in the complex scenario, i.e., the joint punctuation task. we use up to 10 million corpus to train the model. In this joint punctuation task, we use CNN based model as the baseline model, which performs well in the single punctuation task mentioned above. From table II, we find that the proposed model outperforms the CNN based model on "Period", "Comma", "Question", "exclamation" and "Semicolon" with a large gap, i.e., almost up to 30 F value. However, in the experiment of the single punctuation mentioned above, the CNN based model achieves comparable results to the proposed ***SAPR*** model. The reason we conjecture is that the joint punctuation causes much perplexity for the CNN based model predicting the right single label. So in table II, we use "Joint Punctuation" to present all complex successive punctuation and calculate the F value only on the joint punctuation.

We find that when compared with the single punctuation, the F value for joint punctuation is too low. However, our ***SAPR*** model can achieve great performance on either the single punctuation or the joint punctuation. Table II shows that our model achieves at 87.4 F value, which gains significant improvements on the baseline model. The comparison of the CNN model and ***SAPR*** model results reveals that joint punctuation is a much more difficult task. And the traditional model based on the sequence labelling framework can not solve the joint punctuation well.

V. Conclusion and Future Work

In this paper, we propose a novel self-attention based network which views the punctuation restoration as a translation task. With the self-attention based network, we can incorporate the word sequence information and labelling information into the model at the same time. Experimental results show that the proposed model achieves significant improvements than the strong baseline models, especially on the joint punctuation task. We are among the first endeavors to use the translation model to handle the punctuation restoration task, which can be applied to any other sequence labelling task easily. Additionally, we design a constrained decode method to ensure the consistent in the decode time, which would effectively solve the uncertain target word length problem.

VI. Acknowledgements

The research work is supported by the National Key Research and Development Program of China under Grant No. 2017YFB1002102. We would like to thank Xu Shuang for her preparing data used in this work.

References

[1] A. Stolcke, E. Shriberg, R. A. Bates, M. Ostendorf, D. Hakkani, M. Plauche, G. Tür, and Y. Lu, "Automatic detection of sentence boundaries and disfluencies based on recognized words." in *ICSLP*, vol. 2, 1998, pp. 2247–2250.

[2] X. Wang, H. T. Ng, and K. C. Sim, "Dynamic conditional random fields for joint sentence boundary and punctuation prediction," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[3] T. Levy, V. Silber-Varod, and A. Moyal, "The effect of pitch, intensity and pause duration in punctuation detection," in *Electrical & Electronics Engineers in Israel (IEEEI), 2012 IEEE 27th Convention of*. IEEE, 2012, pp. 1–4.

[4] M. Graciarena, E. Shriberg, A. Stolcke, and F. Enos, "Combining prosodic lexical and cepstral systems for deceptive speech detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings*, 2006, pp. I–I.

[5] A. Gravano, M. Jansche, and M. Bacchiani, "Restoring punctuation and capitalization in transcribed speech," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 4741–4744.

[6] D. Zhang, S. Wu, N. Yang, and M. Li, "Punctuation prediction with transition-based parsing," in *Meeting of the Association for Computational Linguistics*, 2013, pp. 752–760.

[7] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," 2016.

[8] W. Lu and H. T. Ng, "Better punctuation prediction with dynamic conditional random fields," in *Proceedings of the 2010 conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2010, pp. 177–186.

[9] N. Ueffing, M. Bisani, and P. Vozila, "Improved models for automatic punctuation prediction for spoken and written text." in *INTERSPEECH*, 2013, pp. 3097–3101.

[10] X. Che, C. Wang, H. Yang, and C. Meinel, "Punctuation prediction for unsegmented transcript based on word vector." in *LREC*, 2016.

[11] W. Gale and S. Parthasarathy, "Experiments in character-level neural network models for punctuation," *Proc. Interspeech 2017*, pp. 2794–2798, 2017.

[12] J. Yi, J. Tao, Z. Wen, and Y. Li, "Distilling knowledge from an ensemble of models for punctuation prediction," in *INTERSPEECH*, 2017, pp. 2779–2783.

[13] O. Tilk and T. Alumäe, "Bidirectional recurrent neural network with attention mechanism for punctuation restoration." in *INTERSPEECH*, 2016, pp. 3047–3051.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[15] K. Ahmed, N. S. Keskar, and R. Socher, "Weighted transformer network for machine translation," 2017.

[16] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "Disan: Directional self-attention network for rnn/cnn-free language understanding," 2017.

[17] O. Tilk and T. Alumäe, "Lstm for punctuation restoration in speech transcripts," in *Sixteenth annual conference of the international speech communication association*, 2015.

[18] D. Griol, J. M. Molina, and Z. Callejas, "Combining speech-based and linguistic classifiers to recognize emotion in user spoken utterances," *Neurocomputing*, 2017.

[19] A. Öktem, M. Farrús, and L. Wanner, "Attentional parallel rnns for generating punctuation in transcribed speech," in *International Conference on Statistical Language and Speech Processing*. Springer, 2017, pp. 131–142.

[20] Z. Yang, W. Chen, F. Wang, and B. Xu, "Improving neural machine translation with conditional sequence generative adversarial nets," 2017.

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[22] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[23] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado *et al.*, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *arXiv preprint arXiv:1611.04558*, 2016.

[24] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling punctuation prediction as machine translation," in *International Workshop on Spoken Language Translation (IWSLT) 2011*, 2011.

[25] E. Cho, J. Niehues, K. Kilgour, and A. Waibel, "Punctuation insertion for real-time spoken language translation," in *Proceedings of the Eleventh International Workshop on Spoken Language Translation*, 2015.

[26] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, and e. a. Sanjay Ghemawat, "Tensorflow: A system for large-scale machine learning," *arXiv preprint arXiv:1605.08695*, 2015.

[27] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[28] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.