PUNCTUATION PREDICTION FOR STREAMING ON-DEVICE SPEECH RECOGNITION

Zhikai Zhou[1], *Tian Tan*[2], *Yanmin Qian*[1†]

[1]MoE Key Lab of Artificial Intelligence, AI Institute
X-LANCE Lab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
[2]AISpeech Ltd, Suzhou, China
{zhikai.zhou,yanminqian}@sjtu.edu.cn,tantiantc@gmail.com

ABSTRACT

Punctuation prediction is essential for automatic speech recognition (ASR). Although many works have been proposed for punctuation prediction, the on-device scenarios are rarely discussed with an end-to-end ASR. The punctuation prediction task is often treated as a post-processing of ASR outputs, but the mismatch between natural language in training input and ASR hypotheses in testing is ignored. Besides, language models built with deep neural networks are too large for edge devices. In this paper, we discuss one-pass models for both ASR and punctuation prediction to replace the conventional two-pass post-processing pipeline. Then the joint ASR-punctuation model is proposed to utilize multi-task learning to decouple the recognition and punctuation on the ASR decoder. Experimental results show that the proposed joint model not only outperforms the traditional post-processing method with limited extra parameters, but also achieves better accuracy in comparison to the direct ASR modeling on transcripts with punctuation.

Index Terms— Streaming speech recognition, edge devices, punctuation prediction, multi-task learning

1. INTRODUCTION

Automatic speech recognition (ASR) systems rarely output any punctuation marks as they are not spoken out, making transcribed text unreadable and resulting in terrible user experiences. Therefore, punctuation prediction is essential for the ASR task. In recent years, on-device speech recognition [1, 2, 3, 4] has become an active research direction due to the needs of privacy protection, low latency and reliability. However, the on-device punctuation prediction was rarely studied.

Punctuation prediction, also known as punctuation restoration, is defined as a sequence tagging task. Many works for punctuation prediction have been proposed previously, which can be categorized based on modality: speech modal [5], text modal [6, 7, 8] and multi-modal containing both [9, 10]. For the text modal, the punctuation prediction task is usually associated with capitalization [11, 12]. Due to the imbalance of data, contrastive learning [13] and focal loss [14] have been adopted for punctuation prediction. Meanwhile, punctuation prediction was treated as a downstream task of unsupervised language models [15] or as an additional task [16] with a pretraining task like the replaced token detection [17]. Also, data augmentation using text-to-speech model [10] has been adopted since the ASR transcripts with punctuation are scarce. Additionally, the real-time punctuation prediction, together with disfluency detection has been studied [18] for streaming scenarios. For multi-modality inputs, fundamental frequency (F0) and energy are the key features extracted from speech data [19].

On the other hand, most works treated punctuation prediction as a post-processing task of ASR outputs. However, the mismatch is usually ignored between the vanilla inputs in the training stage and the ASR hypotheses with errors in the testing stage. The on-device scenario has also been ignored since few works considered the number of model parameters. Hereafter, reducing the number of parameters becomes important for better on-device punctuation prediction.

In this paper, the punctuation prediction is well studied for streaming on-device speech recognition. The joint punctuation-ASR model is proposed to minimize the number of additional parameters for the accurate on-device punctuation prediction. Moreover, the joint model basically resolves the mismatch problem of input in training and inferring.

The main contents of this paper are as follows.

- The joint punctuation-ASR model is proposed for streaming on-device speech recognition, needing only a few additional parameters. As a one-pass model for both ASR and punctuation prediction, the impact of ASR errors is obviously weakened.
- The teacher-forcing decoding method is proposed for the evaluation of the joint punctuation-ASR model on the punctuation prediction task.
- The proposed joint punctuation-ASR model outperforms the post-processing language model in a large margin for punctuation prediction.
- Compared with the direct ASR modeling of transcripts with punctuation marks, the proposed method has few errors on both ASR and punctuation prediction.

2. RELATED WORK

2.1. Triggered Attention for Streaming Speech Recognition

The triggered attention (TA) system [20] is extended from the Transformer, which is composed of an encoder and a decoder. Each module consists of a multi-head self-attention block (MHSA) and several fully-connected layers [21]. The Transformer is trained based on the joint connectionist temporal classification (CTC)/attention framework to achieve fast convergence [22, 23]. The loss function of the

[†] Yanmin Qian is the corresponding author.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

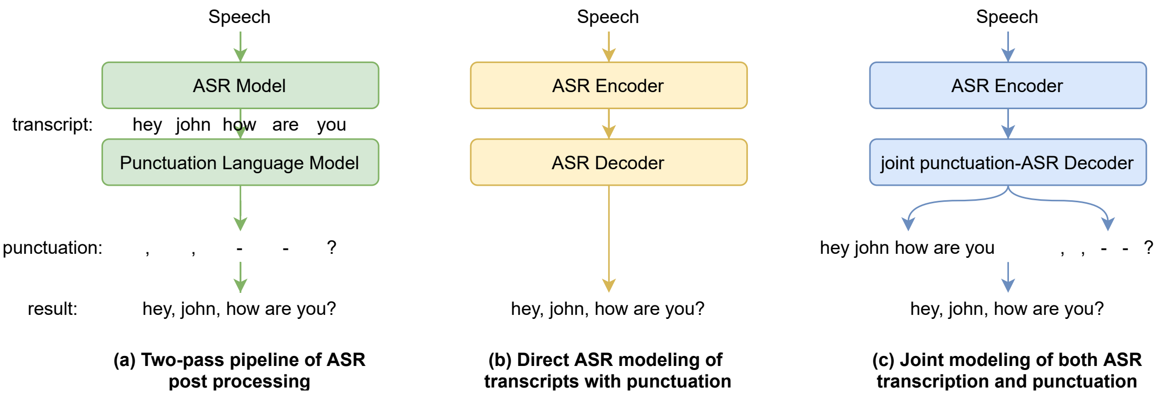

Fig. 1: Three types of punctuation modeling for automatic speech recognition

joint CTC-attention network is defined as:

$$\mathcal{L}_{jca} = \lambda \mathcal{L}_{ctc} + (1 - \lambda)\mathcal{L}_{s2s} \tag{1}$$

where $\mathcal{L}_{ctc}$ and $\mathcal{L}_{s2s}$ are the CTC and sequence-to-sequence (S2S) objectives, respectively. The tunable coefficient $\lambda \in [0, 1]$ is applied to control the contribution of each loss.

For streaming scenarios, the encoder's attention is limited to see only several steps before and after. The input sequence is processed chunk by chunk for testing.

Generally, the trigger mechanism computes spikes of the CTC outputs from the encoder. Then the decoder is triggered whenever the encoder meets a spike, achieving the implicit alignment between the encoder (frame synchronous) and the decoder (label synchronous).

2.2. Punctuation Prediction

With motivation from the superior performance of the pre-trained BERT model on many tasks, the RoBERTa model [24] is adopted and transferred to the punctuation prediction model. For the reasons above, the transformer encoder is used as our architecture.

The inputs of the model here are raw text sequences, while the outputs are punctuation predictions. The punctuation mark of each position in the sequence is defined based on whether the punctuation mark is following the input token. The output labels are punctuation marks and a blank mark.

Subsequently, the model is trained based on the following focal loss $\mathcal{L}_{FL}$ owing to the class imbalance problem:

$$\mathcal{L}_{FL} = -\sum_{k=1}^{N} (1 - p_k)^{\gamma} \hat{y}_k log p_k \tag{2}$$

where N is the total number of categories, p_k is the predicted probability of label k, $\hat{y}_k = 1$ if k is the index of corresponding ground truth class, otherwise $\hat{y}_k = 0$, and γ is the focusing parameter to control the rate at which easy examples are down-weighted.

Meanwhile, to adapt to the streaming scenario, the attention is restricted to observe the previous information for uni-directional modeling. From the view of the whole sequence modeling, the output probability is as follows:

$$P(\mathbf{y}|\mathbf{x}) = \prod_{t} P(y_t|\mathbf{x_{<t}}) \tag{3}$$

where y_t is the prediction according to time step t, $\mathbf{x}_{<t}$ means the whole input sequence before time step t.

3. METHODS FOR PUNCTUATION PREDICTION

3.1. Two-Pass Modeling

Most ASR works consider punctuation prediction as a post-processing task for automatic speech recognition, taking raw text sequence as input and predicting punctuation for each token.

Fig.1 illustrates three types of punctuation modeling for automatic speech recognition. Especially, Fig.1 (a) shows the two-pass pipeline of ASR post-processing for punctuation prediction. The speech is first recognized by an ASR model. Then the punctuation language model (shown in Section 2.2) takes transcripts as input and predicts the punctuation marks for each position. In the end, the transcripts and the punctuation are combined to get the final result.

3.2. One-Pass Modeling

In order to solve the mismatch problem of the punctuation prediction in two-pass modeling and to minimize the number of additional parameters, one-pass modeling methods are proposed.

3.2.1. Direct ASR Modeling

A straightforward method is to utilize transcripts with punctuation marks to train the ASR model directly. The punctuation is treated as a normal token in the model dict for end-to-end ASR. The ASR model directly outputs the final results with punctuation marks, which is shown in Fig.1 (b).

3.2.2. Joint Punctuation-ASR Modeling

The proposed joint model is shown in Fig.1 (c) with two series of output for ASR tokens and punctuation marks, respectively. Also, the model is trained using the multi-task learning framework. The parameters of the proposed model can be directly transferred from an existed ASR model. Since only the decoder is utilized for actual token prediction in the trigger attention mechanism, the joint model is trained using the following multi-task loss $\mathcal{L}_{mtl}$ without the CTC loss:

$$\mathcal{L}_{mtl} = \mathcal{L}_{s2s} + \alpha \mathcal{L}_{FL} \tag{4}$$

where the $\mathcal{L}_{FL}$ is from Equation 2 for punctuation prediction, $\mathcal{L}_{s2s}$ is the S2S loss from Equation 1 for the ASR decoder, and the tunable α controls the weight for punctuation prediction.

For the joint punctuation-ASR model, Fig. 2 shows different methods for the joint modeling for punctuation and ASR tasks,

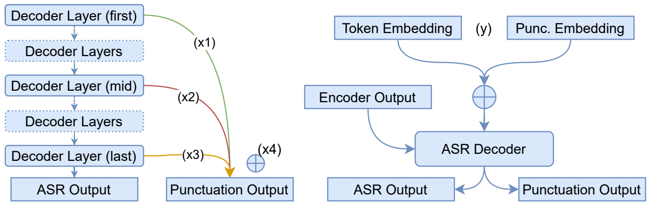

Fig. 2: Different methods for joint punctuation-ASR model

which are marked as x1, x2, x3, x4, and y. For x1, x2, x3, and x4, the decoder takes previous tokens and hidden representations as inputs, then the features from different decoder layers are exploited using a linear projection layer for punctuation prediction. For example, method x1 uses the output of the first layer. Also, features from decoder layers can be summed together and exploited for punctuation prediction. Importantly, the method y not only takes the previous token as input but previous punctuation marks are also taken. Moreover, the embeddings of both tokens and punctuation marks are summed as the input of the ASR decoder. In decoding phases, we conduct beam search on pure ASR results and take argmax results on punctuation.

3.3. Teacher-forcing Decoding Scheme

An important problem for evaluating one-pass pipeline punctuation prediction is that the errors from ASR and punctuation prediction are combined. Whether the ASR or the punctuation prediction is wrong cannot be determined from any punctuation-related error due to the presence of insertion and deletion errors. Therefore, for the results of the one-pass models, we cannot directly calculate the F_1-score.

However, for the auto-regressive decoder, the idea of teacher-forcing training is referred to evaluate the punctuation prediction. For the one-pass models, we can compute the posterior of punctuation marks at time step s_t as follows:

$$P(s_t|x) = P(s_t|h, \hat{y}_{<t}, \hat{s}_{<t}) \quad (5)$$

$$P(h|x) = Encoder(x) \quad (6)$$

while x is the speech feature sequence, h is the encoded hidden representation, and $\hat{y}_{<t}$ and $\hat{s}_{<t}$ denotes previous ground truth transcripts and symbols, respectively. The teacher-forcing scheme utilizes the ground truth labels as the decoder input, which circumvents errors of the ASR.

4. EXPERIMENTS

4.1. Dataset

We evaluate the ASR and punctuation prediction performance on an in-house Chinese dataset. The 3000 hours of Chinese spoken utterances with both transcripts and punctuation are adopted. They are randomly partitioned into the train and development sets into 90%-10% split. We have three test sets: Indoor, Meeting, and Mobile. Indoor is from daily dialogues under the noisy indoor scenarios; Meeting comes from the conversation and discussion from the conference; Mobile is the test set of recordings from social and game mobile apps. The size of Indoor, Meeting, and Mobile test sets are 10 hours, 5 hours, and 16 hours, respectively. The punctuation annotations consist of five kinds of symbols: comma, period, question mark, enumeration comma, and blank. In order to evaluate the performance of both ASR and punctuation prediction, we report the whole sequence token error rate (TER), character-only error rate (CER), and F_1-score (F_1) using the proposed teacher-forcing decoding scheme. We don't compute the F_1-score of blank symbols.

4.2. Streaming On-Device Speech Recognition

4.2.1. Streaming ASR Setup

We follow the basic setup of the transformer model and the input in the literature [25], including a transformer with 12 encoder and 6 decoder layers, and the multi-head self-attention with the dimension of 64 with 8 heads. The SpecAugment [26] is conducted on speech features. The 6979 Chinese characters are adopted as the modeling units while they exist more than ten times in the training transcripts. For streaming scenarios, the features are chunked by 20 encoder steps, actually being 80 frames due to the four times of sub-sampling. For each chunk, the hidden representation is stored in a cache with no more than 4 seconds of information to save the computing cost. Meanwhile, the CTC output is computed based on the current chunk to count the number of the spikes. The decoder takes the hidden representation in the cache and the previous token sequence as inputs to predict the next token.

4.2.2. Deploying on Edge Devices

In order to reduce memory consumption and accelerate the inference to meet real-time requirements, the parameters of the model are quantized from 32-bit floating-point into 8-bit fixed-point. The dynamic quantization is adopted in our implementation. The quantization procedure is expressed as follows:

$$\theta_Q = \frac{\theta - z}{s} \quad (7)$$

where θ is the model parameter, z is the zero_input and s is the scale. For dynamic quantization, the parameters are quantized ahead, while the activations are dynamically quantized during inference. We leverage the ONNX [27] format and runtime for deployment. On ARM architectures, the quantization achieves three times of speedup in comparison to the floating-point execution.

4.3. Two-Pass Pipeline with Punctuation Model

For the transformer language model, we follow the setup in literature [24] while using different numbers of layers. The vocabulary uses the same 6979 Chinese characters as in Section 4.2.1. The models take RoBERTa as initialization and are trained on a large-scale conversation text corpus with 10 million long sentences. Then they are finetuned on the transcripts of 3000 hours ASR dataset.

For evaluation metrics, the whole sequence token error rate (TER) reflects the overall performance of the final system containing both the ASR model and the punctuation model. In addition, the F_1-score is an important metric for punctuation prediction in communities.

Model	#params	Indoor TER/F_1	Mobile TER/F_1	Meeting TER/F_1
Trans-2L	9.88M	15.93/86.80	27.94/70.69	28.89/72.64
Trans-4L	16.19M	15.88/87.28	27.84/71.48	28.76/74.09
Trans-6L	22.49M	15.72/87.59	27.73/71.79	28.64/74.15
ASR	72.6M	13.49 (CER)	24.89 (CER)	25.91 (CER)

Table 1: Performance Comparison of the Two-Pass Strategy with Punctuation Models

Model	α	#Ext par.	Indoor	Mobile	Meeting	Average
			TER/CER/F_1	TER/CER/F_1	TER/CER/F_1	TER/CER/F_1
ASR + Trans-6L	-	22.49M	15.72/13.49/87.59	27.73/24.89/71.79	28.64/25.91/74.15	24.03/21.43/77.84
ASR with Punc	-	11.3K	15.49/14.45/**92.02**	31.73/28.87/71.66	31.88/29.95/78.06	26.37/24.42/80.58
Joint Model -x3	1.0	2.0K	14.62/**13.19**/91.01	24.27/21.39/72.38	**27.76/25.33**/78.82	22.22/19.97/80.74
Joint Model -x3	2.0	2.0K	**14.51**/13.20/91.45	**23.66/20.60**/71.70	28.46/26.15/78.29	**22.21/19.98**/80.48
Joint Model -x3	5.0	2.0K	14.53/13.36/92.00	24.48/21.59/**72.17**	28.77/26.53/**79.11**	22.59/20.49/**81.09**
Joint Model -x1	2.0	2.0K	39.51/17.54/50.51	57.92/35.58/35.21	46.35/37.74/53.51	47.93/30.29/46.41
Joint Model -x2	2.0	2.0K	20.10/13.68/79.84	35.44/25.99/58.57	34.68/27.77/69.74	30.07/22.48/69.38
Joint Model -x3	2.0	2.0K	**14.51/13.20**/91.45	**23.66/20.60**/71.70	28.46/26.15/78.29	**22.21/19.98**/80.48
Joint Model -x4	2.0	2.0K	14.61/13.23/91.17	25.31/22.40/70.91	**28.29/25.83**/77.72	22.74/20.49/79.93
Joint Model -y	2.0	4.0K	14.56/13.24/91.20	24.82/21.95/**72.30**	29.23/27.01/**78.46**	22.87/20.73/**80.65**

Table 2: Performance comparison of different strategies for both ASR and punctuation prediction. ASR+Trans-6L: The two-pass pipeline using punctuation language models. ASR with punc: The one-pass direct ASR modeling on transcripts with punctuation. Joint Model utilizes feature from which output of the decoder layer: x1: 1st, x2: 3rd, x3: Last, x4: Sum of all, y: Last, but feed punctuation result to the input.

In Table.1, the whole sequence TER and the F_1 score on three test sets for punctuation of the two-pass models are displayed. "#params" means the number of parameters of the model. The raw text sequence character error rate (CER) of the streaming vanilla ASR model is shown in the last row. The CER is computed using the ASR hypotheses and the transcripts without punctuation marks. The uni-directional language models for punctuation prediction are evaluated, using both ASR hypotheses from the model in the last row and ground truth text sequences. "Trans-xL" means we adopt the transformer encoder with x layers ($x \in \{2, 4, 6\}$). The "TER" in this table means the whole sequence (both characters and punctuation) token error rate, computed from the reference texts with punctuation and the outputs of the language model using ASR hypotheses. Meanwhile, the "F_1" is computed from the reference texts with punctuation and the outputs of the language model using ground truth raw texts. In addition, the "F_1" is averaged through four kinds of punctuation marks described in Section 4.1. The results show that with the decrease of parameters, the performance does not get worse apparently.

4.4. One-Pass ASR-Punctuation Models

For one-pass ASR-punctuation models, the metrics contain the TER and F_1 mentioned above together with the raw CER to evaluate the ASR performance. To get the F_1 score, the teacher-forcing decoding scheme is utilized to obtain a correct result.

The direct ASR modeling of transcripts with punctuation follows the same setup in Section 4.2.1 while using different modeling units containing 6979 Chinese characters and punctuation marks. For the joint punctuation-ASR model, we have five styles or methods for joint modeling. For methods x1 to x3, we utilize the output of the first layer, the third layer, and the last layer from the decoder, respectively. Then a linear layer is used to project the embedding dimension 512 to the punctuation output dimension 5. For method x4, the outputs of all layers are firstly summed together, and then the linear layer of the same setup mentioned before is used. Besides, An extra embedding layer is introduced for method y in comparison to the setup of method x3. These embeddings of punctuation and characters are summed together.

Table 2 exhibits the performance of all strategies on three test sets for both ASR and punctuation prediction. "#Ext par." indicates the number of extra parameters for punctuation prediction. For the two-pass model, this means the parameters of the language model. In contrast, for one-pass models, this means the parameters of the extra module, such as the linear projection layer for punctuation. Comparing the first two rows with each other in Table.2, we find that the performance for the raw ASR in the first row is degraded for direct ASR modeling because we force the ASR to model the unspoken tokens, leading to more errors. Conversely, the punctuation task performs better due to the utilization of both speech and previous contexts.

We tune the coefficient α in Equation 4 to control the training balance between the ASR task and the punctuation prediction based on method x3 for the joint Punc-ASR modeling. For the third to the fifth row in Table.2, the best performance for both TER and CER is achieved in case of $\alpha = 2.0$. All three methods have better performance than the two-pass model in the first row and the direct ASR model in the second row, while requiring much fewer extra parameters. It is observed that the introduction of the focal loss for punctuation prediction exerts the regularization effect on the ASR task, which makes the decoder learn better in comparison to the raw ASR in the first row. For the direct ASR modeling results in the second row, the proposed methods achieve better accuracy on both ASR and punctuation prediction.

Then, we adopted $\alpha = 2.0$ for all the subsequent experiments, and tried all the methods mentioned in Section 3.2.2. It is shown in Table.2 that the projection of shallow layers in methods x1 and x2 performs worse than that from other methods. But methods x3,x4, and y perform much better than the two-pass and direct ASR models. Moreover, x3 achieves the best performance both in TER and CER, with a similar F_1 score to the best model. Compared with the two-pass model and the direct ASR model, the joint model x3 achieves 7.6% and 15.7% relative TER reduction, 6.8% and 18.2% relative CER reduction, respectively, leading to much fewer errors for punctuation prediction with very limited extra parameters.

5. CONCLUSIONS

In this paper, we decouple the ASR and punctuation prediction task while using the same ASR decoder. The joint punctuation-ASR model is proposed for streaming on-device speech recognition, introducing only a few additional parameters. The teacher-forcing decoding scheme is proposed for the evaluation of one-pass models. The proposed joint model achieves much better performance on both ASR and punctuation tasks over the two-pass pipeline and the direct ASR model while leveraging limited extra parameters.

6. ACKNOWLEDGEMENTS

This work was supported by the China NSFC projects (No. 62122050 and No. 62071288), and Shanghai Municipal Science and Technology Major Project (No. 2021SHZDZX0102).

7. REFERENCES

[1] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6381–6385.

[2] K. Kim, K. Lee, D. Gowda, J. Park, S. Kim, S. Jin, Y.-Y. Lee, J. Yeo, D. Kim, S. Jung *et al.*, "Attention based on-device streaming speech recognition with large speech corpus," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 956–963.

[3] W. Li, J. Qin, C.-C. Chiu, R. Pang, and Y. He, "Parallel rescoring with transformer for streaming on-device speech recognition," *arXiv preprint arXiv:2008.13093*, 2020.

[4] I. McGraw, R. Prabhavalkar, R. Alvarez, M. G. Arenas, K. Rao, D. Rybach, O. Alsharif, H. Sak, A. Gruenstein, F. Beaufays *et al.*, "Personalized speech recognition on mobile devices," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5955–5959.

[5] F. Batista, H. Moniz, I. Trancoso, and N. Mamede, "Bilingual experiments on automatic recovery of capitalization and punctuation of automatic speech transcripts," *IEEE transactions on audio, speech, and language processing*, vol. 20, no. 2, pp. 474–485, 2012.

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

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

[8] M. Courtland, A. Faulkner, and G. McElvain, "Efficient automatic punctuation restoration using bidirectional transformers with robust inference," in *Proceedings of the 17th International Conference on Spoken Language Translation*, 2020, pp. 272–279.

[9] O. Klejch, P. Bell, and S. Renals, "Sequence-to-sequence models for punctuated transcription combining lexical and acoustic features," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5700–5704.

[10] D. Soboleva, O. Skopek, M. Šajgalík, V. Cărbune, F. Weissenberger, J. Proskurnia, B. Prisacari, D. Valcarce, J. Lu, R. Prabhavalkar *et al.*, "Replacing human audio with synthetic audio for on-device unspoken punctuation prediction," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7653–7657.

[11] B. Nguyen, V. B. H. Nguyen *et al.*, "Fast and accurate capitalization and punctuation for automatic speech recognition using transformer and chunk merging," in *2019 22nd Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA)*. IEEE, 2019, pp. 1–5.

[12] M. S. S. R. K. Dixit and S. B. K. Kirchhoff, "Robust prediction of punctuation and truecasing for medical asr," *ACL 2020*, p. 53, 2020.

[13] Q. Huang, T. Ko, H. L. Tang, X. Liu, and B. Wu, "Token-level supervised contrastive learning for punctuation restoration," *arXiv preprint arXiv:2107.09099*, 2021.

[14] J. Yi, J. Tao, Z. Tian, Y. Bai, and C. Fan, "Focal loss for punctuation prediction." in *INTERSPEECH*, 2020, pp. 721–725.

[15] K. Makhija, T.-N. Ho, and E.-S. Chng, "Transfer learning for punctuation prediction," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2019, pp. 268–273.

[16] M. Hentschel, E. Tsunoo, and T. Okuda, "Making punctuation restoration robust and fast with multi-task learning and knowledge distillation," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7773–7777.

[17] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," *arXiv preprint arXiv:2003.10555*, 2020.

[18] Q. Chen, M. Chen, B. Li, and W. Wang, "Controllable time-delay transformer for real-time punctuation prediction and disfluency detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8069–8073.

[19] G. Szaszák and M. A. Tündik, "Leveraging a character, word and prosody triplet for an asr error robust and agglutination friendly punctuation approach." in *INTERSPEECH*, 2019, pp. 2988–2992.

[20] N. Moritz, T. Hori, and J. Le Roux, "Triggered attention for end-to-end speech recognition," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5666–5670.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[22] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2017, pp. 4835–4839.

[23] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," in *Proc. Interspeech 2017*, 2017, pp. 949–953.

[24] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[25] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on transformer vs RNN in speech applications," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 449–456.

[26] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[27] "Open neural network exchange (onnx)," Microsoft, 2021. [Online]. Available: https://github.com/microsoft/onnxruntime