Boosting Punctuation Restoration
with Data Generation and Reinforcement Learning

Viet Dac Lai[1] , *Abel Salinas*[3], *Hao Tan*[2], *Trung Bui*[2], *Quan Tran*[2], *Seunghyun Yoon*[2],
Hanieh Deilamsalehy[2], *Franck Dernoncourt*[2], *Thien Huu Nguyen*[1]

[1]Dept. of Computer Science, University of Oregon, USA
[2]Adobe Research, USA, [3]University of Southern California, USA
{vietl,thien}@cs.uoregon.edu asalinas@isi.edu
{haotan,bui,qtran,syoon,deilamsa,franck.dernoncourt}@adobe.com

Abstract

Punctuation restoration is an important task in automatic speech recognition (ASR) which aim to restore the syntactic structure of generated ASR texts to improve readability. While punctuated texts are abundant from written documents, the discrepancy between written punctuated texts and ASR texts limits the usability of written texts in training punctuation restoration systems for ASR texts. This paper proposes a reinforcement learning method to exploit in-topic written texts and recent advances in large pre-trained generative language models to bridge this gap. The experiments show that our method achieves state-of-the-art performance on the ASR test set on two benchmark datasets for punctuation restoration. The source code of this work is publicly accessible at https://github.com/laiviet/pr-rl.

Index Terms: punctuation restoration, reinforcement learning.

1. Introduction

Automatic Speech Recognition (ASR) is a key component in processing audio materials such as audio translation and voice assistant [1], and speech information extraction [2]. Typical ASR systems produce chunks of transcription without any text structures such as sentence and phrase boundaries [3]. As a result, it lowers the readability of the generated ASR texts [3] and severely affects the performance of systems for downstream tasks over this type of text, e.g., information extraction [4]. To address this issue, the Punctuation Restoration (PR) task has been added to the ASR systems [5] to improve the text readability and the performance of downstream tasks for ASR-generated texts such as question answering [6], chitchat detection [7], and tutorial recommendation [8]. The most recent successful work for PR was all built on top of transformer-based PLMs such as BERT [9] and ELECTRA [10].

Despite such progress, lacking domain-specific training data is still a major obstacle that hinders the research and development of PR systems for real-world applications [11]. We identify two factors accounting for this issue. First, speech topics involve a unique set of keywords as well as slang in spoken languages. The ASR system and PR system without topic knowledge can be severely affected by the shift of topics in the source audio. Second, unlike other tasks where the unlabeled data is created by humans, the input of PR is generated by an ASR system. This creates a unique dependency that must be addressed by the PR model. Consequently, creating cost-effective datasets for a wide range of domains for PR is highly challenging.

Moreover, naive adoption of available punctuated data is problematic. While large-scale punctuated texts corpora are available, they are mostly written texts (REF texts), which are substantially well-punctuated. In contrast, ASR-generated texts (ASR texts) inherit a substantial amount of noise from both spoken language (e.g., verbal pauses) and the transcription process (e.g., word errors). Accordingly, prior studies have shown that a PR model that was trained on REF texts performed poorly on real-world ASR texts [12]. In other words, directly using readily available written texts does not help to improve the PR model.

To overcome these issues, we introduce a novel data generation method to automatically generate large-scale, high-quality labeled data for PR. In particular, instead of manual annotation, we employ a pre-trained language model, namely GPT2 [13], to create synthetic labeled data for PR because generative models like GPT2 can generate punctuated texts that can be converted to labeled data for PR easily. Since the GPT2 model was trained on written texts across diverse topics, this leads to two issues that need to be addressed.

First, the topics in the generated texts are unconstrained, which is suboptimal for some specific applications, such as gaming livestreaming. As such, we propose a method to control the topic of the generated texts. Instead of unconditional text generation, we feed the GPT2 model with an in-topic seed text, which was sampled from an in-topic unsupervised source. Hence, we encourage the GPT2 model to generate more texts within the initial topic. As a result, we can leverage GPT2's knowledge to obtain unlimited in-topic labeled texts for PR.

Second, the disconnection of the GPT2 model and the target PR model might cause a discrepancy between GPT2-generated texts and the target PR text. Therefore, to improve the quality of the GPT2-generated data for PR, we propose to further finetune the GPT2 model in parallel with the training of the PR model to generate optimal customized texts for PR. Particularly, we propose a meta-learning framework to consider the GPT2 model as a meta-parameter for the training of the PR model, in which the GPT2 model will be fine-tuned based on the performance of the PR model on the development set. A trivial solution is reinforcement learning, where the reward can be calculated directly from the evaluation metrics of the PR model on the development set, e.g., the F1-score. However, obtaining a reliable, fast reward is challenging due to either the small scale of the evaluation or the computational cost of the evaluation that has to be done at every single iteration. To alleviate this issue, we propose a novel reward function that relies on the gradients of the PR model obtained from the generated texts and the development set. Intuitively, a generated sample should have a higher reward if the PR model's gradients derived from the sample follows the PR expected gradients derived from the development set. Toward this end, in each iteration, after generating synthetic PR data, we compute an average gradient of the PR model over the generated data for each training example. Then, we compute

another average gradient of the PR model over a sampled subset of the development set. Finally, the reward for each generated sample is computed using the cosine similarity score between the two gradients. We evaluate the effectiveness of the proposed methods on two benchmark datasets for PR. The experiments show that our model outperforms the strongest baseline on both datasets.

2. Related Work

Early PR studies employed syntactic features and prosodic features[14] to train graphical models such as HMM and CRF [5]. Recent models for PR employed artificial neural networks to model the PR problem as a sequence-to-sequence problem using various network architectures such as convolutional neural network [15], recurrent neural network [5, 16], and transformer [12]. Pretrained language models stand at the core of the recent PR models. There have been variants of pre-trained language models used for PR such as BERT [17], RoBERTa [12, 18], ELECTRA [19, 20], XLM-RoBERTa [21], and funnel-transformer [22]. Recent advance in training and preprocessing leads to many training techniques such as data augmentation [12], adversarial training [23], multitask learning [24, 19], self-training [20], two-stage training [17], and contrastive learning [25]. External knowledge was also incorporated into the PR model including external punctuated data [17], syntactic features [22] and acoustic features [26].

3. Proposed Approach

3.1. Problem Setting

Similar to prior studies, we model the PR task as a word-level sequence labeling problem. Given a text input sequence $X = \{w_1, w_2, \cdots, w_N\}$ where N is the number of words in the whole sequence, the input X is encoded into vector space using a large language model, parameterized as f_θ, as $H = \{h_1, h_2, \cdots, h_N\}$. The ground truth corresponding to the input sequence is $Y = \{y_1, y_2, \cdots, y_N\}$ where y_i belongs to a predefined list of punctuation marks. The model's prediction is formalized as $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_N\}$. The PR model, parameterized as $\mathcal{M}^\theta$, is trained using cross-entropy loss:

$$\mathcal{L}_{CE} = -\frac{1}{N} y_i \log \hat{y}_i$$

We propose a reinforcement learning framework to leverage a generative PLM to adaptively generate PR data for training the PR model. In particular, the learning process involves two models: a PR model $\mathcal{M}^\theta$ and a GPT2 model $\mathcal{M}^\omega$. It first generates in-topic punctuated text from some seed texts derived from the in-topic unsupervised text. Then, the PR model $\mathcal{M}^\theta$ is trained on both the generated data and human-annotated data. Afterward, the GPT2 model $\mathcal{M}^\omega$ is finetuned based on the feedback from the PR model to further improve the effectiveness of the generated data. This is achieved by a reinforcement learning algorithm that exploits the agreement between generated data and the development set of the human-annotated data to form the reward function. Algorithm 1 presents the detail of the proposed method.

3.2. Data Augmentation

Training/testing data discrepancy is a crucial problem in the punctuation restoration task. The training data that are obtained from written text, however, does not reflect the noise in the ac-

	Model	Comma	Period	Question	Overall
REF	ELECTRA-base [110M]	72.7	89.7	**89.7**	81.1
	+ Multitask	76.2	89.0	86.0	82.6
	RoBERTa-large [335M]	76.3	88.6	81.9	82.4
	+ Augmentation	76.7	88.9	87.8	82.9
	funnel-transformer-xlarge [400M]	78.8	88.9	86.6	83.7
	+ POS Fusion + SBS	78.4	89.8	89.4	84.3
	Electra-large [335M]	79.0	90.0	86.0	84.4
	+ Discriminative Self-Training	80.1	90.4	86.0	**85.2**
	DeBERTa-large [304M]	78.7	89.6	87.5	84.1
	+ RL (Ours)	**80.1**	**90.4**	82.8	85.1
ASR	ELECTRA-base [110M]	58.4	81.4	64.0	68.9
	+ Multitask	62.0	82.9	67.6	71.9
	funnel-transformer-xlarge [400M]	62.3	81.5	61.9	70.8
	+ POS Fusion + SBS	63.2	82.8	66.7	72.6
	RoBERTa-large [355M]	61.8	81.9	58.1	71.3
	+ Augmentation	66.3	82.3	63.4	74.0
	DeBERTa-large [304M]	62.0	82.5	68.4	71.4
	+ RL (Ours)	**69.2**	**84.7**	**72.7**	**77.0**

Table 1: *Punctuation prediction performance comparison (F1 score) on the IWSLT corpus. The Upper half of the table reports the performance on reference texts, while the lower half reports the performance on the ASR texts. Note that ELECTRA-large + Disc Self-Training model [20] did not report performance on ASR text.*

tual spoken text that is transcribed by an ASR system. As such, to introduce noise to the text, we augmented the input text using three strategies: *duplication, alternation, deletion* with an augmentation probability of $\alpha_1, \alpha_2, \alpha_3$ similar to prior work [12]. To fit the very long input sequence into a large language model, the input sequence must be split into shorter segments of the same size. Due to the randomness of the chunking, the predictions of the edge tokens (head and tail of the chunk) might be severely affected due to the lack of preceding or following contexts. To overcome this, we feed additional preceding and following words of a chunk to help the large language model better encodes the sequence for the PR task, especially for predicting the chunk's beginning and ending words. In particular, we concatenate C preceding and C following words to the input sequence, if they are available, resulting in the input sequence $X_C = \{C, X, C\}$ fed to the PR model. We do not predict the labels for these additional tokens to avoid prediction conflict with the preceding and tailing chunks, as well as to prevent the lack of context to recur.

3.3. Data Generation

Due to the limited annotated in-topic data for PR, we proposed a more feasible method to generate an unlimited amount of data for PR using a generative language model, named GPT2. As GPT2 was trained on a massive amount of unsupervised learning text across many topics, it can generate a long piece of text given just a short seed prompt, which controls the topic of the generated text through the seed text given to GPT2. To do that, we obtained the transcripts of the TED-talk from 2013 to 2017 (separated from the IWSLT corpus which covers talks before 2012); then we used this as our unsupervised in-topic corpus for text generation. For the BehancePR corpus, we use the unsupervised text in the development set as the in-topic seed.

In particular, in each iteration, a batch of semi-annotated data $\mathcal{B}^{gen}$ is generated by the GPT2 model $\mathcal{M}^\omega_{t-1}$ using an in-topic seed $\mathcal{B}^{seed}$. Another batch $\mathcal{B}^{train}$ is sampled from the

Algorithm 1 Reinforcement Learning for PR

Require: $\mathcal{M}^\omega, \mathcal{M}^\theta, f_\theta, f_\omega$
Require: $\mathcal{D}^{unsup}$
Require: $\mathcal{D}^{train}, \mathcal{D}^{dev}$
  for $t <$ max_iteration **do**
    $\mathcal{B}^{seed} \leftarrow sample(\mathcal{D}^{unsup})$
    $\mathcal{B}^{gen} \leftarrow \mathcal{M}^\omega_{t-1}(\mathcal{B}^{seed})$ ▷ Generate data
    $\mathcal{B}^{train} \leftarrow sample(\mathcal{D}^{train})$
    $\theta_t \leftarrow update(\theta_{t-1}, \nabla f_\theta(\mathcal{B}^{gen} \cup \mathcal{B}^{train})$ ▷ Update PR model
    $\mathcal{B}^{dev} \leftarrow sample(\mathcal{D}^{dev})$
    $grad^{dev} \leftarrow \nabla f_\theta(\mathcal{B}^{dev})$
    $grad^{gen} \leftarrow \nabla f_\theta(\mathcal{B}^{gen})$
    $r = grad^{dev} \times grad^{gen}$ ▷ Compute reward
    $\nabla_\omega = \sum_{b_i \in \mathcal{B}^{gen}} r_i \nabla f_\omega(b_i)$
    $\omega_t \leftarrow update(\omega_{t-1}, \nabla_\omega)$ ▷ Update GPT2 model
  end for

---

original PR training data $\mathcal{D}^{train}$. Finally, the PR model $\mathcal{M}^\theta$ is trained on the combined batch of these two batches.

3.4. Reinforcement Learning:

The GPT2 model is helpful in generating well-punctuated in-topic data. However, as the generation is done independently from the PR model, the generated data inherits the written language style from the GPT2 model's memory. As a result, the generated data is not optimal for the PR task, as the ultimate goal of PR is to be used for spoken language. As such, it is necessary for the PR model to give feedback to the GPT2 model so that the GPT2 model can be finetuned in parallel with the training of the PR model. Expectedly, the guidance from the PR model can make the GPT2 model generate more relevant text.

One trivial way to measure the effectiveness of the generated data is the performance of the PR model (e.g., overall F1-score) over the development set. However, as the label in a PR dataset is highly imbalanced, using a discrete measure like F1-score might lead to a high variance reward, hence, inaccurate estimation. Moreover, we aim to train the GPT2 such that the model can learn to generate a sample $\mathcal{B}^{gen}$ that resembles the language style in the development set $\mathcal{D}^{dev}$. Intuitively, the generated text should be similar to the spoken human language if the gradient updates of the model trained on $\mathcal{B}^{gen}$ and $\mathcal{D}^{dev}$ are aligned. Formally, the reward r_i for each batch of generated texts $\mathcal{B}^{gen}$ is computed as follows:

$$r_i = \nabla_\theta \mathcal{L}(\mathcal{B}^{gen}_i; \theta_{t-1}) \cdot \sum_{\mathcal{B}_j \in \mathcal{D}^{dev}} \frac{\nabla_\theta \mathcal{L}(\mathcal{B}_j; \theta_{t-1}))}{|\mathcal{D}^{dev}|} \quad (1)$$

where $\mathcal{L}(\mathcal{B}; \theta_{t-1})$ is the cross-entropy of training the PR model $\mathcal{M}^\theta_{t-1}$ on the sample $\mathcal{B}$ and $\cdot$ denotes dot product.

Finally, the GPT2 model is trained to maximize negative log-likelihood:

$$\mathcal{L}_G = - \sum_{\mathcal{B}_i \in \mathcal{B}^{gen}} r_i log P(B_i) \quad (2)$$

4. Experiments

Settings: In this paper, we evaluate our proposed model on two available English datasets that have been used in previous studies. **IWSLT** is the benchmark dataset for the PR task in English. It annotates three prominent punctuation marks:

Model	P	R	F1
RoBERTA-large	62.0	61.4	61.7
+ Augmentation	63.8	60.7	62.2
+ CRF	62.2	63.5	62.9
+ CRF + Augmentation	61.1	62.8	62.0
DeBERTa-large	63.7	64.8	64.2
+ RL (Ours)	**64.1**	**66.2**	**65.2**

Table 2: *Performances on the BehancePR test set. Note that [11] did not report the breakdown performance for each type.*

PERIOD, COMMA, QUESTION. The IWSLT corpus contains texts derived from TED Talks, which are mainly monologues. The testing set of this corpus contains both reference text (REF), which is well-written text, and transcribed text (ASR) with manually inserted punctuation. Whereas the training set consists of only REF text. The training, development, and test sets contain approximately 2.1M, 300K, and 12K words, respectively. **BehancePR** is a human-annotated dataset for livestreaming videos. It features multiple speakers as well as interaction with a large number of audiences. BehancePR corpus contains only ASR text. The training/development/testing sets contain approximately 1.2M, 34K, and 44K words, respectively. The models are evaluated using the standard precision, recall, and F1-score (micro).

Hyperparameters: In this paper, each input word is tokenized using the word-piece tokenizer provided in the PLM. The representation of the first word-piece is collected as the input of the classifier head, which is a fully connected layer, to predict the punctuation. We employed the DeBERTa-large PLM [27] as the encoder of the PR model. The hidden states of the top 8 layers are used as the representation of a token, searched from a pool of $\{1,4,8,12\}$ layers. The GPT2-medium is used to generate the text. The seed texts for the GPT2 model contain 64 consecutive words randomly sampled from these pools. Both models are trained using the Adam optimizer with a learning rate in $\{2e\text{-}5, 5e\text{-}5\}$. The augmentation ratios $\alpha_1, \alpha_2, \alpha_3$ are set to 5%, similar to [12]. We concatenate $C = 20$ context words to the head and tail of each chunk. Due to the high cost of evaluating the PR model on the whole development set, in each iteration, we only sample a subset $|B_j| = 16$ chunks from $\mathcal{D}_{dev}$ to compute the reward.

4.1. IWSLT corpus

Baselines: We compared our model with the state-of-the-art PR models: **RoBERTa-large+Augmentation** model employs a RoBERTa-large PLM [12]. The input data is augmented using three augmentation strategies: insertion, substitution, and deletion. **ELECTRA-base+Multitask** [19] is finetuned using additional augmentation detection loss and knowledge distillation loss. **ELECTRA-large+Discriminative Self-Training** [20] was self-trained with a discriminator to detect human-annotated data and pseudo-machine-labeled data. **Funnel-transformer-xlarge+POSFusion** [22] incorporates additional part-of-speech features from an external neural-network based POS tagger.

Results: Table 1 compares the examined models' performance on both the REF test set and the ASR test set. The performance on the REF test set shows us the performance in case the ASR text is close to the written text, while the ASR test shows the actual performance on ASR text.

On the REF test set, ELECTRA-large is the best model among the five examined PLMs with an F1 score of 84.4%,

Model	P	R	F1
RoBERTa-large	66.5	76.7	71.3
+ Augmentation	72.0	76.2	74.0
+ GPT + RL	73.3	76.7	75.0
DeBERTa-large	66.1	77.6	71.4
+ Augmentation	73.0	77.1	75.0
+ GPT	74.9	76.3	75.6
+ RL (Full model)	**74.6**	**79.4**	**77.0**
DeBERTa-large + GPT + RL			
+ PR pretraining (1 epoch)	74.5	78.5	76.4
+ PR pretraining (2 epochs)	74.2	78.3	76.2
+ GPT2 pretraining (1 epoch)	75.7	77.0	76.3
+ GPT2 pretraining (2 epochs)	74.5	77.2	75.8

Table 3: *Performances on the IWSLT ASR test set.*

and it is closely followed by DeBERTa-large (0.3% lower). These models leave a large margin to the smaller models such as ELECTRA-base (approx. 3% lower). Comparing the full models, our DeBERTA-large + RL model gains 1% over the DeBERTa-large model, achieving 85.1%. This performance is on par with the ELECTRA-large + Discriminative Self-Training model with a mere margin of 0.1%.

For ASR text, comparing the full models, our DeBERTa-large + RL model (77% in terms of overall F1) outperforms all the other models at a large margin of 3% to the highest competitor, RoBERTa-large + Augmentation, with $p < 0.01$. Moreover, without additional training signals or external features, the DeBERTa-large model yields similar performance to other PLMs (e.g., RoBERTa-large and funnel-transformer-xlarge). Furthermore, our proposed model outperforms the other models on all three punctuation marks with a consistently large margin ranging from 1.8% to 5.1%, compared to the next highest. These results clearly show the robustness of our proposed RL method to boost the performance of real-world ASR data significantly. The improvement suggests that the RL method has provided helpful training examples to help the model bridge the gap between the REF text and the ASR text in the training and testing data, respectively.

4.2. BehancePR corpus

Baselines: We compare our models with the state-of-the-art models that have been evaluated on this corpus. These models include the **RoBERTa-large** model and its variants with **Data Augmentation** and **Conditional Random Field** [12].

Results: First, we found that data augmentation does not improve the performance of the model trained on the BehancePR dataset. The reason is that the BehancePR dataset's training and testing data are all ASR texts, which is different from the IWSLT corpus in which the training texts are REF texts, and the testing texts are ASR texts. As such, introducing data augmentation skewed the distribution of training and testing data in the BehancePR corpus. Hence, hurting the model's performance. Table 2 presents the overall performance of our proposed models on the BehancePR corpus. The DeBERTa-large outperforms the current state-of-the-art RoBERTa-large+CRF model (62.2% versus 62.9%). Furthermore, the DeBERTa-large + RL improves the F1 score from 64.2% to 65.2% (+1.0) (statistically significant with $p < 0.01$). This again shows the effectiveness of the proposed reinforcement learning methods.

4.3. Ablation study

We perform an ablation study to examine the contribution of each component of the model on the IWSLT ASR test set as shown in Table 3 (Rows 1-7). Adding the augmentation to the DeBERTa-large model boosts the performance from 71.4% to 75.0% (**+3.6%**), while *GPT* improves the F1 score from 75.0% to 75.6% (**+0.6%**). Finally, when we add *RL*, the F1 score jumps from 75.6% to 77.0%. These demonstrate that all the proposed components contribute to the improvement. However, data augmentation and RL contribute largely to the performance gain on the IWSLT ASR test set. Finally, to further show the effectiveness of the *RL*, we add it to the RoBERTa-large+Augmentation, resulting in an increase of 1% in the F1 score. This experiment shows that our RL method is model-agnostic that can be applied to any PR model.

The PR model and the GPT2 model could be finetuned/pre-trained with different strategies. To examine whether finetuned or pre-trained model before the reinforcement learning could further improve the performance of the model. We used the configuration of the full model with GPT2 and RL. However, for the PR model, we trained the PR alone with the same training data for 1 and 2 epochs. Similarly, we pre-trained the GPT2 model on the unsupervised text derived from the training set for the same epochs. Table 3 (Rows 8-12) reports the performance of these runs. As can be seen from the performance, training/finetuning the model using only PR or GPT2 data significantly hurts the performance of the model. In particular, pre-training a single epoch on PR or GPT2 reduced the performance by 0.4% to 0.7%, respectively. Further training the model for one more epoch decreased the performance by 0.4% to 0.5%, respectively.

5. Conclusion

This paper focuses on generating helpful training data for the punctuation restoration task, especially for real-world ASR texts. We devise a reinforcement learning method to use the GPT2 model to generate additional data to train the punctuation restoration model. This method allows the GPT2 model to learn from real-world ASR text to generate more helpful training examples based on gradient feedback from the PR model. Our model improves PR performance on real-world ASR tests on IWSLT and BehancePR (+3% and +2.3%, respectively). In the future, we would like to extend this research with more advanced gradient feedback to improve the generated data.

6. References

[1] C. Federmann and W. D. Lewis, "Microsoft speech language translation (MSLT) corpus: The IWSLT 2016 release for English, French and German," in *Proceedings of IWSLT*, 2016.

[2] S. Cho, F. Dernoncourt, T. Ganter, T. Bui, N. Lipka, W. Chang, H. Jin, J. Brandt, H. Foroosh, and F. Liu, "StreamHover: Livestream transcript summarization and annotation," in *Proceedings of EMNLP*, 2021, pp. 6457–6474. [Online]. Available: https://aclanthology.org/2021.emnlp-main.520

[3] D. Jones, F. Wolf, E. Gibson, E. Williams, E. Fedorenko, D. Reynolds, and M. Zissman, "Measuring the readability of automatic speech-to-text transcripts•," 2003.

[4] F. Alam, B. Magnini, and R. Zanoli, "Comparing named entity recognition on transcriptions and written texts," in *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*. Springer, 2015, pp. 71–89.

[5] O. Tilk and T. Alumäe, "Lstm for punctuation restoration in speech transcripts," in *Sixteenth annual conference of the international speech communication association*, 2015. [Online]. Available: https://www.isca-speech.org/archive_v0/interspeech_2015/papers/i15_0683.pdf

[6] A. Pouran Ben Veyseh, V. Lai, F. Dernoncourt, and T. Nguyen, "BehanceQA: A new dataset for identifying question-answer pairs in video transcripts," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, Jun. 2022, pp. 7321–7327. [Online]. Available: https://aclanthology.org/2022.lrec-1.796

[7] V. Lai, A. Pouran Ben Veyseh, F. Dernoncourt, and T. Nguyen, "BehanceCC: A ChitChat detection dataset for livestreaming video transcripts," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, Jun. 2022, pp. 7284–7290. [Online]. Available: https://aclanthology.org/2022.lrec-1.791

[8] A. P. B. Veyseh, F. Dernoncourt, and T. H. Nguyen, "Tutorial recommendation for livestream videos using discourse-level consistency and ontology-based filtering," *Proceedings of the Video Transcript Understanding Workshop at AAAI 2022*, 2022.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[10] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," in *ICLR*, 2020. [Online]. Available: https://openreview.net/pdf?id=r1xMH1BtvB

[11] V. Lai, A. Pouran Ben Veyseh, F. Dernoncourt, and T. Nguyen, "BehancePR: A punctuation restoration dataset for livestreaming video transcript," in *Findings of NAACL-HLT*, 2022, pp. 1943–1951. [Online]. Available: https://aclanthology.org/2022.findings-naacl.149

[12] T. Alam, A. Khan, and F. Alam, "Punctuation restoration using transformer models for high-and low-resource languages," in *Proceedings of W-NUT*, 2020, pp. 132–142.

[13] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[14] G. Szaszák and M. A. Tündik, "Leveraging a character, word and prosody triplet for an asr error robust and agglutination friendly punctuation approach." in *Proc. of INTERSPEECH*, 2019, pp. 2988–2992. [Online]. Available: https://www.isca-speech.org/archive_v0/Interspeech_2019/pdfs/2132.pdf

[15] X. Che, C. Wang, H. Yang, and C. Meinel, "Punctuation prediction for unsegmented transcript based on word vector," in *Proceedings of LREC*, 2016, pp. 654–658. [Online]. Available: https://aclanthology.org/L16-1103

[16] S. Kim, "Deep recurrent neural networks with layer-wise multi-head attentions for punctuation restoration," in *ICASSP*, 2019, pp. 7280–7284. [Online]. Available: https://ieeexplore.ieee.org/document/8682418

[17] X.-Y. Fu, C. Chen, M. T. R. Laskar, S. Bhushan, and S. Corston-Oliver, "Improving punctuation restoration for speech transcripts via external data," in *Proceedings of W-NUT 2021*, 2021, pp. 168–174.

[18] M. Courtland, A. Faulkner, and G. McElvain, "Efficient automatic punctuation restoration using bidirectional transformers with robust inference," in *Proceedings of IWSLT*, 2020, pp. 272–279. [Online]. Available: https://aclanthology.org/2020.iwslt-1.33

[19] M. Hentschel, E. Tsunoo, and T. Okuda, "Making punctuation restoration robust and fast with multi-task learning and knowledge distillation," in *ICASSP*, 2021, pp. 7773–7777. [Online]. Available: https://ieeexplore.ieee.org/document/9414518

[20] Q. Chen, W. Wang, M. Chen, and Q. Zhang, "Discriminative Self-Training for Punctuation Prediction," in *INTERSPEECH*, 2021, pp. 771–775. [Online]. Available: https://www.isca-speech.org/archive/pdfs/interspeech_2021/chen21d_interspeech.pdf

[21] V. Chordia, "PunKtuator: A multilingual punctuation restoration system for spoken and written text," in *Proceedings of EACL*, 2021, pp. 312–320. [Online]. Available: https://aclanthology.org/2021.eacl-demos.37

[22] N. Shi, W. Wang, B. Wang, J. Li, X. Liu, and Z. Lin, "Incorporating external pos tagger for punctuation restoration," *Proc. of INTERSPEECH*, pp. 1987–1991, 2021. [Online]. Available: https://www.isca-speech.org/archive/pdfs/interspeech_2021/shi21_interspeech.pdf

[23] J. Yi, J. Tao, Y. Bai, Z. Tian, and C. Fan, "Adversarial transfer learning for punctuation restoration," *arXiv preprint arXiv:2004.00248*, 2020.

[24] B. Lin and L. Wang, "Joint prediction of punctuation and disfluency in speech transcripts." in *Proc. of INTERSPEECH*, 2020, pp. 716–720. [Online]. Available: https://www.isca-speech.org/archive_v0/Interspeech_2020/pdfs/1277.pdf

[25] Q. Huang, T. Ko, H. L. Tang, X. Liu, and B. Wu, "Token-Level Supervised Contrastive Learning for Punctuation Restoration," in *Proc. of INTERSPEECH*, 2021, pp. 2012–2016. [Online]. Available: https://www.isca-speech.org/archive/pdfs/interspeech_2021/huang21g_interspeech.pdf

[26] Y. Zhu, L. Wu, S. Cheng, and M. Wang, "Unified multimodal punctuation restoration framework for mixed-modality corpus," in *ICASSP*, 2022, pp. 7272–7276. [Online]. Available: https://ieeexplore.ieee.org/document/9747131

[27] P. He, X. Liu, J. Gao, and W. Chen, "Deberta: Decoding-enhanced bert with disentangled attention," in *ICLR*, 2021. [Online]. Available: https://openreview.net/forum?id=XPZIaotutsD