Discriminative Self-training for Punctuation Prediction

Qian Chen, Wen Wang, Mengzhe Chen, Qinglin Zhang

Speech Lab, Alibaba Group

{tanqing.cq, w.wang, mengzhe.cmz, qinglin.zql}@alibaba-inc.com

Abstract

Punctuation prediction for automatic speech recognition (ASR) output transcripts plays a crucial role for improving the readability of the ASR transcripts and for improving the performance of downstream natural language processing applications. However, achieving good performance on punctuation prediction often requires large amounts of labeled speech transcripts, which is expensive and laborious. In this paper, we propose a Discriminative Self-Training approach with weighted loss and discriminative label smoothing to exploit unlabeled speech transcripts. Experimental results on the English IWSLT2011 benchmark test set and an internal Chinese spoken language dataset demonstrate that the proposed approach achieves significant improvement on punctuation prediction accuracy over strong baselines including BERT, RoBERTa, and ELECTRA models. The proposed Discriminative Self-Training approach outperforms the vanilla self-training approach. We establish a new state-of-the-art (SOTA) on the IWSLT2011 test set, outperforming the current SOTA model by 1.3% absolute gain on F_1.

Index Terms: punctuation prediction, self-training, label smoothing, Transformer, BERT

1. Introduction

Spoken language transcripts generated by automatic speech recognition (ASR) systems usually have no punctuation marks. However, many downstream applications, such as machine translation and dialogue systems, are usually trained on well-formed text with proper punctuation marks. Hence, there is a significant mismatch between the training corpora and the actual ASR transcript input to these applications, causing dramatic performance degradation. In addition, lack of punctuation marks reduces the readability of speech transcripts. Consequently, punctuation prediction has become a crucial post-processing task for speech transcripts.

A critical challenge for punctuation prediction is that to achieve good performance on actual ASR output, many existing approaches require large amounts of human-annotated spoken language data. However, acquiring labeled spoken language data is a costly process. There have been several approaches to alleviate this issue through transfer learning, such as using large amounts of written language data with punctuation marks for self-supervised training for punctuation prediction [1, 2, 3]. However, there exists significant discrepancy between written language data and spoken language data. A large amount of spoken language data is available for training ASR models; yet many spoken language datasets, such as LibriSpeech [4], do not have transcripts with manually labeled punctuation marks. In this work, we propose a Discriminative Self-Training (denoted Disc-ST) approach for exploiting unlabeled speech transcripts without punctuation marks and demonstrate its effectiveness in improving strong baselines for punctuation prediction.

Our contributions can be summarized as follows:

1) To the best of our knowledge, this paper is the first to explore self-training for punctuation prediction. We exploit large-scale unlabeled spoken language data without punctuation, such as transcripts used for training ASR systems, through self-training to improve strong baseline models based on BERT, RoBERTa, and ELECTRA.

2) We propose a Discriminative Self-Training approach using weighted loss and discriminative label smoothing when training on combined human-labeled and pseudo-labeled data.

3) Experimental results on the English IWSLT2011 benchmark test set and an internal Chinese spoken language dataset demonstrate that our approach achieves significant improvement on punctuation prediction accuracy over strong baselines including BERT, RoBERTa, and ELECTRA models. The proposed discriminative self-training approach outperforms the vanilla self-training approach. We establish a new state-of-the-art (SOTA) on the IWSLT2011 test set, outperforming the current SOTA model by 1.3% absolute on F_1.

2. Related Work

Previous punctuation prediction models can be categorized into three major categories. The first category views punctuation prediction as hidden inter-word event detection, using models such as n-gram language models [5] and Hidden Markov Models (HMMs) [6, 7]. The second category treats punctuation prediction as sequence labeling by assigning a punctuation mark to each word using conditional random fields (CRFs) [7, 8, 9], convolutional neural networks (CNNs) [10, 11], recurrent neural networks (RNNs) [11, 12] and their variants [13, 14]. The third category uses sequence-to-sequence modeling in which the source is unpunctuated text and the target is punctuated text [15] or sequences of punctuation marks [16, 17].

Some previous punctuation prediction approaches explored lexical information and acoustic/prosodic information separately and in combination. An important observation from these works is that information from text plays a much more crucial role in punctuation prediction than prosodic information; still, adding prosodic information to text-based modeling may achieve performance improvement [18, 19]. In this work, we focus on improving text-based punctuation prediction models over the current SOTA which only explores text information.

Self-supervised learning can help reduce the required amount of labeling. A model pre-trained on unlabeled data through self-supervised learning can be fine-tuned on smaller amount of labeled data. Recently there has been much research exploring this pretraining-finetuning framework, built upon transformer-based pre-trained language models [2, 20, 3, 21, 22]. Different from these previous studies, our work explores self-training(ST) [23] in finetuning transformer-based pre-trained models for punctuation prediction.

In the standard procedure of ST, a teacher model trained with labeled data is used to generate pseudo labels on unlabeled data. Then the data with pseudo labels are combined with la-

beled data to train a student model. ST has proven effective on many tasks, such as image classification [24], word sense disambiguation [23], disfluency detection [25], parsing [26], machine translation [27], and text summarization [27]. Different from the tasks on which ST was applied in previous works, this work investigates the efficacy of applying ST on punctuation prediction. On top of the vanilla ST approach, we explore discriminative weighted loss and labeling smoothing to improve the effectiveness of self-training.

3. Proposed Approach

3.1. Model Architecture

Our model treats punctuation prediction as a sequence labeling task. The input are transcripts without punctuation. For languages without explicit word boundaries such as Chinese, the input is segmented into words. Our model predicts whether there is a specific punctuation mark after each word. The word tokens are processed with WordPiece tokenization and the output are fed into a Transformer encoder [28]. The final hidden state of the encoder corresponding to the last sub-token is used as the input to the softmax classifier for punctuation prediction.

3.2. Discriminative Self-Training

Figure 1 provides an overview of the proposed discriminative self-training approach for punctuation prediction, given large-scale written language data, human-labeled spoken language data with punctuation labels $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ and unlabeled spoken language data $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_m$.

First, we pre-train a language model on large-scale well-formed text corpora by self-supervised tasks, such as masked language modeling (MLM) and next sentence prediction (NSP), to obtain deep bidirectional language representations.

We then initialize a teacher model θ^t from the pre-trained language model and train it on spoken language data with human-labeled punctuation marks through minimizing the cross-entropy loss ℓ:

$$\ell = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i, \theta^t)) \tag{1}$$

where f denotes the classifier. Next, we infer pseudo labels on clean unlabeled spoken language data as follows:

$$\tilde{y}_i = f(\tilde{x}_i, \theta^t)), \forall i = 1, \ldots, m \tag{2}$$

Then, we initialize a student model θ^s from pre-trained language models and train it by minimizing the cross-entropy loss on the combination of human-labeled spoken language data and pseudo-labeled spoken language data.

In this paper, we propose a self-training approach employing weighted loss and label smoothing in a discriminative way when training on the combination of human-labeled data and pseudo-labeled data, denoted discriminative self-training (**Disc-ST**). Intuitively, the pseudo-labeled data has more noise than human-labeled data. Thus, we use different weights to combine the loss on human-labeled data and pseudo-labeled data (denoted **weighted loss**), as follows:

$$\ell = \sum_{i=1}^{n} \ell(y_i, f(x_i, \theta^s)) + \alpha \sum_{i=1}^{m} \ell(\tilde{y}_i, f(\tilde{x}_i, \theta^s)) \tag{3}$$

where α is a weight factor. Meanwhile, we explore one of the output regularizers, label smoothing [29], to deal with training

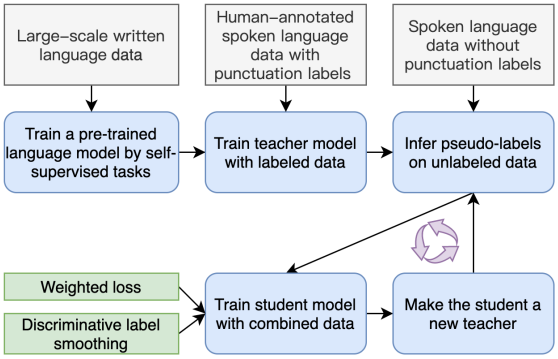

Figure 1: *Illustration of the proposed discriminative self-training approach for punctuation prediction.*

with noisy labels. In future work, we plan to explore other output regularizers such as focal loss [30] and bootstrapping loss [31]. The standard cross-entropy loss is calculated as

$$\ell = -\sum_{i=1}^{K} y_i log(p_i) = -y^T log(p) \tag{4}$$

where $y = \text{One-hot}(i)$, $p = [p_1, p_2, \ldots, p_K]$, K is the number of classes, and i is the label index. For label smoothing, we replace y with y_{LS} as

$$y_{LS} = (1 - \beta)y + \beta/K \tag{5}$$

Similar to the motivation for weighted loss, considering the different noise levels in human-labeled data and pseudo-labeled data, we use label smoothing discriminatively, i.e., we use different factors (β_1 and β_2) for human-labeled data and pseudo-labeled data. α, β_1 and β_2 are hyperparameters optimized on the validation set. After Disc-ST, we put the student back as the teacher, and iterate this process until the performance converges. The student model achieving the best performance on the human-labeled validation set is chosen as our final model.

3.3. Double-Overlap Sliding Window Decoding Strategy

When evaluating punctuation prediction performance on a test set, it is infeasible to decode a very long utterance without any segmentation all at once, since it will cause unacceptable latency and out-of-memory issues. Previous work studied different decoding strategies to segment long sequences, including overlapping windows [32], streaming input scheme [33], overlapped-chunk split and merging algorithm [34], and a fast decoding strategy [1]. We extend the overlapped-chunk split and merging algorithm [34] to improve punctuation prediction accuracy. As showed in Figure 2, we use a sliding window with both left and right overlapped context, and only keep predictions of the model where there is enough context information on both sides. The step size equals the window size minus the sum of the left overlap size and the right overlap size. In our experiments, we observe that the left context is more important than the right context for punctuation prediction accuracy. The main difference from the original overlapped-chunk split and merging algorithm is that our step size can be tuned by optimizing the left and right overlap sizes (as two hyperparameters) based on the performance on a validation set; whereas their step size is not tuned and is set as half of the window size.

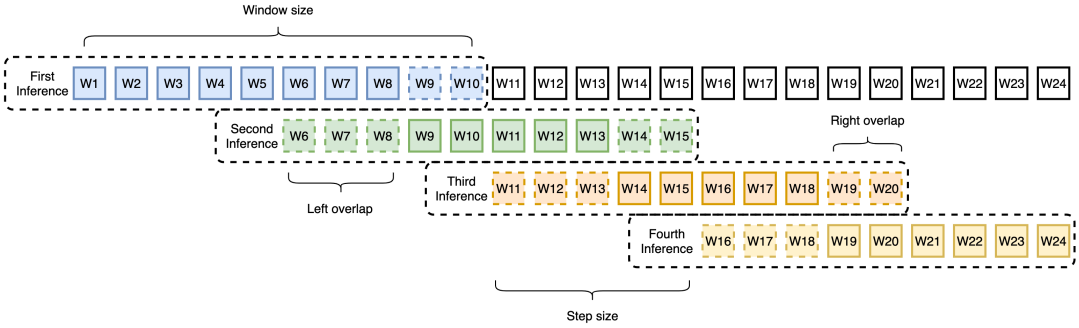

Figure 2: *Illustration of double-overlap sliding window decoding strategy.*

4. Experiments

4.1. Datasets

We evaluate the punctuation prediction accuracy on the English IWSLT2011 benchmark dataset and an internal Chinese spoken language dataset. The IWSLT2011 benchmark contains three types of punctuation marks (comma, period, and question mark). We use the same data organization and same tokenized data as Che et al. [10][1] used. We also compare the proposed approach with previous methods on a large internal Chinese dataset [1]. We use Jieba[2] for Chinese word segmentation. The punctuation annotations consist of four types of punctuation marks (comma, period, question mark, and enumeration comma). For unlabeled spoken language data, we use LibriSpeech [4], Fisher Speech Transcripts Part 1 and Part 2 [35] for the English dataset and use internal speech transcripts without punctuation for the Chinese dataset. Data statistics are summarized in Table 1. We evaluate the punctuation prediction performance using token-based precision (P), recall (R), F_1-score (F_1), following previous works [12].

Dataset	Split	#Words	#Punctuation
IWSLT2011	Train	2M	301K
	Unlabeled	30M	-
	Dev	296K	43K
	Test	13K	2K
Chinese	Train	5M	745K
	Unlabeled	68M	-
	Dev	132K	18K
	Test	93K	15K

Table 1: *Statistics of train, unlabeled speech data, dev, and test sets for IWSLT2011 and internal Chinese datasets.*

4.2. Training Details

For all experiments on English IWSLT2011, we use the "BERT-base, Uncased" model (12 layers, 768 hidden units, 12 heads, 110M parameters) [36][3] and "ELECTRA-large" model (24 layers, 1024 hidden units, 16 heads, 335M parameters) [37][4]. For IWSLT2011, we train 3 epochs, and use batch size 32 and initial learning rate 5e-5 for BERT-base; and batch size 16, initial learning rate 2e-5 for ELECTRA-large. The window size is set to 120, the right overlap size is set to 15, and the left overlap size is set to 35 for BERT-base and 40 for ELECTRA-large. For the Chinese dataset, we use the "RoBERTa-wwm-base, Chinese" models (12 layers, 768 hidden units, 12 heads, 110M parameters) [38][5]. We use batch size 64, initial learning rate 2e-5, and train 2 epochs. The number of self-training iterations is 1 in all of our experiments.

4.3. Performance on English IWSLT2011

We compare the proposed approach with previous models on English IWSLT2011. The first group of models and results listed in Table 2 is cited from previous works. T-LSTM [12] used uni-directional LSTM and T-BRNN-pre [13] used bidirectional RNN with attention. BLSTM-CRF and Teacher-Ensemble are the best single and ensemble models in [14], respectively. DRNN-LWMA-pre [39] used a deep recurrent neural network architecture with layer-wise multi-head attentions. Self-attention-word-speech [16] used a full sequence Transformer encoder-decoder model with pretrained word2vec and speech2vec embeddings. CT-Transformer [1] used a controllable time-delay Transformer and utilized well-formed text corpora to pretrain. SAPR [40] used a Transformer encoder-decoder which views the punctuation restoration as a translation task. BERT-base+Adversarial [3] used BERT-base with adversarial multi-task learning by adding an extra POS tagging task. BERT-large+Transfer [2] used BERT-large with LSTM-CRF. BERT-base+FocalLoss [20] improved BERT-base with focal loss to alleviate the class imbalance problem. RoBERTa-large+augmentation [21] used RoBERTa-large with data augmentation, including insertion, substitution, and deletion. The current SOTA (RoBERTa-base) [22] used RoBERTa-base with aggregating predictions across multiple context windows.

The second group shows the experimental results from this work (using the decoding strategy in Section 3.3). We fine-tune BERT-base and ELECTRA-large models with the human-labeled training set and observe that ELECTRA-large outperforms the current SOTA (RoBERTa-base) (F_1 84.4 versus 83.9) and significantly outperforms BERT-base (F_1 84.4 versus 77.4). Vanilla self-training (denoted vanilla ST) on BERT-base improves F_1 from 77.4 to 79.0 (**+1.6**), and on ELECTRA-large improves F_1 from 84.4 to 84.7 (**+0.3**), demonstrating that vanilla ST obtains significant gains over the strong baselines. Replacing vanilla ST with Discriminative Self-Training (Disc-ST) on BERT-base achieves a further improvement on F_1 from 79.0 to

[1] https://github.com/IsaacChanghau/neural_sequence_labeling
[2] https://github.com/fxsjy/jieba
[3] https://github.com/google-research/bert
[4] https://github.com/google-research/electra
[5] https://github.com/ymcui/Chinese-BERT-wwm

Model	Comma			Period			Question			Overall		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
T-LSTM [12]	49.6	41.4	45.1	60.2	53.4	56.6	57.1	43.5	49.4	55.0	47.2	50.8
T-BRNN-pre [13]	65.5	47.1	54.8	73.3	72.5	72.9	70.7	63.0	66.7	70.0	59.7	64.4
BLSTM-CRF [14]	58.9	59.1	59.0	68.9	72.1	70.5	71.8	60.6	65.7	66.5	63.9	65.1
Teacher-Ensemble [14]	66.2	59.9	62.9	75.1	73.7	74.4	72.3	63.8	67.8	71.2	65.8	68.4
DRNN-LWMA-pre [39]	62.9	60.8	61.9	77.3	73.7	75.5	69.6	69.6	69.6	69.9	67.2	68.6
Self-attention-word-speech [16]	67.4	61.1	64.1	82.5	77.4	79.9	80.1	70.2	74.8	76.7	69.6	72.9
CT-Transformer [1]	68.8	69.8	69.3	78.4	82.1	80.2	76.0	82.6	79.2	73.7	76.0	74.9
SAPR [40]	57.2	50.8	55.9	96.7	97.3	96.8	70.6	69.2	70.3	78.2	74.4	77.4
BERT-base+Adversarial [3]	76.2	71.2	73.6	87.3	81.1	84.1	79.1	72.7	75.8	80.9	75.0	77.8
BERT-large+Transfer [2]	70.8	74.3	72.5	84.9	83.3	84.1	82.7	93.5	87.8	79.5	83.7	81.4
BERT-base+FocalLoss [20]	74.4	77.1	75.7	87.9	88.2	88.1	74.2	88.5	80.7	78.8	84.6	81.6
RoBERTa-large+augmentation [21]	76.8	76.6	76.7	88.6	89.2	88.9	82.7	93.5	87.8	82.6	83.1	82.9
RoBERTa-base [22]	76.9	75.4	76.2	86.1	89.3	87.7	88.9	87.0	87.9	84.0	83.9	**83.9**
BERT-base	70.6	72.8	71.6	82.7	83.6	83.2	71.9	89.1	79.6	76.3	78.4	77.4
BERT-base+Vanilla ST	73.0	74.5	73.7	82.9	85.5	84.2	74.5	89.1	81.2	77.8	80.2	79.0
BERT-base+Disc-ST	73.7	74.7	74.2	83.2	86.6	84.9	75.9	89.1	82.0	78.4	80.8	**79.6**
ELECTRA-large	76.3	81.9	79.0	89.3	90.8	90.0	79.6	93.5	86.0	82.4	86.5	84.4
ELECTRA-large+Vanilla ST	77.4	82.0	79.6	89.5	90.5	90.0	81.1	93.5	86.9	83.1	86.4	84.7
ELECTRA-large+Disc-ST	78.0	82.4	80.1	89.9	90.8	90.4	79.6	93.5	86.0	83.6	86.7	**85.2**

Table 2: *Punctuation prediction results in terms of P(%), R(%), F1(%) on the English IWSLT2011 test set.*

Model	Comma			Period			Question			Enum. Comma			Overall		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
BLSTM [1]	58.9	43.9	50.3	59.7	58.1	58.9	77.0	58.8	66.7	59.8	16.5	25.9	60.2	48.8	53.9
Full-Transformer [1]	61.9	50.7	55.8	60.5	64.7	62.5	74.2	68.6	71.3	64.5	30.9	41.8	62.1	55.9	58.8
CT-Transformer [1]	60.8	53.5	56.9	63.8	59.7	61.7	76.3	63.0	69.0	63.4	25.2	36.1	62.7	55.3	58.8
RoBERTa-wwm-base	63.8	51.0	56.7	65.6	60.8	63.1	71.7	75.8	73.7	43.8	47.4	45.5	64.2	55.6	59.6
RoBERTa-wwm-base + Disc-ST	62.6	53.0	57.4	64.0	64.0	64.0	73.2	71.4	72.3	49.8	34.6	40.8	63.5	57.3	**60.2**

Table 3: *Punctuation prediction results in terms of P(%), R(%), F1(%) on the internal Chinese test set.*

79.6 (**+0.6**), a statistically significant improvement with $p < 0.01$; for ELECTRA-large, Disc-ST improves F_1 from 84.7 by vanilla ST to 85.2 (**+0.5**), a statistically significant improvement with $p < 0.01$. Our ELECTRA-large+Disc-ST outperforms the current SOTA (RoBERTa-base) by **1.3** absolute F_1 gain.

4.4. Performance on Chinese Dataset

We also compare the proposed approach with previous models on the Chinese dataset. The first group of models and results in Table 3 is cited from previous works. The second group shows the experimental results from this work. To compare with previous methods in [1], we use the same fast decoding strategy [1] with a low frame rate 3 and the number of look-ahead words after end-of-sentence mark as 6. The RoBERTa-wwm-base model significantly outperforms the current SOTA CT-Transformer (F_1 59.6 versus 58.8). Furthermore, Discriminative Self-Training improves F_1 from 59.6 to 60.2 (**+0.6**), a statistically significant improvement with $p < 0.01$.

4.5. Ablation Study

We conduct ablation study to understand the contributions of component algorithms to the overall system performance on the IWSLT2011 test set, as shown in Table 4. When we add vanilla ST to BERT-base, F_1 improves from 77.4 to 79.0 (**+1.6**). When we further add weighed loss, F_1 improves from 79.0 to 79.3 (**+0.3**). When we further add equal label smoothing (LS) for both human-labeled data and pseudo-labeled data, F_1 improves from 79.3 to 79.4 (**+0.1**). When we replace equal LS with discriminative LS, F_1 improves from 79.3 to 79.6 (**+0.3**). These results demonstrate that both weighted loss and discriminative LS in discriminative self-training contribute to the improvement over vanilla self-training.

Model	Overall		
	P	R	F_1
BERT-base	76.3	78.4	77.4
+ Vanilla ST	77.8	80.2	79.0
+ weighted loss	78.0	80.7	79.3
+ label smoothing (LS)	78.4	80.4	79.4
+ discriminative LS	78.4	80.8	79.6

Table 4: *Ablation study of Discriminative Self-training.*

5. Conclusions

We propose a Discriminative Self-Training approach with weighted loss and discriminative label smoothing. Experimental results show that the proposed approach achieves significant improvement on punctuation prediction over strong baselines including BERT, RoBERTa, and ELECTRA models. The proposed Discriminative Self-Training approach outperforms vanilla self-training. Our approach outperforms the current SOTA on the English IWSLT2011 benchmark and an internal Chinese dataset. Future work includes investigating other output regularizers and the efficacy of self-training in cross-domain and cross-lingual applications.

6. References

[1] Q. Chen, M. Chen, B. Li, and W. Wang, "Controllable time-delay transformer for real-time punctuation prediction and disfluency detection," in *ICASSP*, 2020, pp. 8069–8073.

[2] K. Makhija, T. Ho, and E. S. Chng, "Transfer learning for punctuation prediction," in *APSIPA*, 2019, pp. 268–273.

[3] J. Yi, J. Tao, Y. Bai, Z. Tian, and C. Fan, "Adversarial transfer learning for punctuation restoration," *CoRR*, vol. abs/2004.00248, 2020.

[4] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP*, 2015, pp. 5206–5210.

[5] D. Beeferman, A. L. Berger, and J. D. Lafferty, "Cyberpunc: a lightweight punctuation annotation system for speech," in *ICASSP*, 1998, pp. 689–692.

[6] H. Christensen, Y. Gotoh, and S. Renals, "Punctuation annotation using statistical prosody models," in *ITRW*, 2001.

[7] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. P. Harper, "Enriching speech recognition with automatic detection of sentence boundaries and disfluencies," *IEEE TASLP*, vol. 14, no. 5, pp. 1526–1540, 2006.

[8] W. Lu and H. T. Ng, "Better punctuation prediction with dynamic conditional random fields," in *EMNLP*, 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] P. Zelasko, P. Szymanski, J. Mizgajski, A. Szymczak, Y. Carmiel, and N. Dehak, "Punctuation prediction model for conversational speech," in *Interspeech*, 2018, pp. 2633–2637.

[12] O. Tilk and T. Alumäe, "LSTM for punctuation restoration in speech transcripts," in *INTERSPEECH*, 2015, pp. 683–687. [Online]. Available: http://www.isca-speech.org/archive/interspeech_2015/i15_0683.html

[13] Tilk and Alumäe, "Bidirectional recurrent neural network with attention mechanism for punctuation restoration," in *Interspeech*, 2016, pp. 3047–3051. [Online]. Available: https://doi.org/10.21437/Interspeech.2016-1517

[14] 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. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/1079.html

[15] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling punctuation prediction as machine translation," in *IWSLT*, 2011, pp. 238–245. [Online]. Available: http://www.isca-speech.org/archive/iwslt_11/sltb_238.html

[16] J. Yi and J. Tao, "Self-attention based model for punctuation prediction using word and speech embeddings," in *ICASSP*, 2019, pp. 7270–7274. [Online]. Available: https://doi.org/10.1109/ICASSP.2019.8682260

[17] O. Klejch, P. Bell, and S. Renals, "Punctuated transcription of multi-genre broadcasts using acoustic and lexical approaches," in *SLT*, 2016, pp. 433–440. [Online]. Available: https://doi.org/10.1109/SLT.2016.7846300

[18] J. Huang and G. Zweig, "Maximum entropy model for punctuation annotation from speech," in *INTERSPEECH*, 2002.

[19] Y. Liu, N. V. Chawla, M. P. Harper, E. Shriberg, and A. Stolcke, "A study in machine learning from imbalanced data for sentence boundary detection in speech," *Comput. Speech Lang.*, vol. 20, no. 4, pp. 468–494, 2006.

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

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

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

[23] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *ACL*, 1995, pp. 189–196. [Online]. Available: https://www.aclweb.org/anthology/P95-1026/

[24] Q. Xie, M. Luong, E. H. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *CVPR*, 2020, pp. 10 684–10 695. [Online]. Available: https://doi.org/10.1109/CVPR42600.2020.01070

[25] S. Wang, Z. Wang, W. Che, and T. Liu, "Combining self-training and self-supervised learning for unsupervised disfluency detection," in *EMNLP*, 2020.

[26] D. McClosky, E. Charniak, and M. Johnson, "Effective self-training for parsing," in *HLT-NAACL*, 2006.

[27] J. He, J. Gu, J. Shen, and M. Ranzato, "Revisiting self-training for neural sequence generation," in *ICLR*, 2020.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[29] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016, pp. 2818–2826.

[30] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, 2020.

[31] S. E. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," in *ICLR*, 2015.

[32] E. Cho, J. Niehues, and A. Waibel, "Segmentation and punctuation prediction in speech language translation using a monolingual translation system," in *IWSLT*, 2012, pp. 252–259. [Online]. Available: http://www.isca-speech.org/archive/iwslt_12/sltc_252.html

[33] E. Cho, J. Niehues, K. Kilgour, and A. Waibel, "Punctuation insertion for real-time spoken language translation," in *IWSLT*, 2015.

[34] B. Nguyen, V. B. H. Nguyen, H. Nguyen, P. N. Phuong, T. Nguyen, Q. T. Do, and L. C. Mai, "Fast and accurate capitalization and punctuation for automatic speech recognition using transformer and chunk merging," *CoRR*, vol. abs/1908.02404, 2019.

[35] C. Cieri, D. Graff, O. Kimball, D. Miller, and K. Walker, "Fisher english training speech part 1 transcripts ldc2004t19 and part 2 transcripts ldc2005t19," 2005.

[36] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019, pp. 4171–4186.

[37] K. Clark, M. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: pre-training text encoders as discriminators rather than generators," in *ICLR*, 2020.

[38] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang, and G. Hu, "Revisiting pre-trained models for chinese natural language processing," in *Findings of EMNLP*, 2020.

[39] S. Kim, "Deep recurrent neural networks with layer-wise multi-head attentions for punctuation restoration," in *ICASSP*, 2019, pp. 7280–7284.

[40] F. Wang, W. Chen, Z. Yang, and B. Xu, "Self-attention based network for punctuation restoration," in *ICPR*, 2018, pp. 2803–2808.