Incorporating External POS Tagger for Punctuation Restoration

Ning Shi¹, Wei Wang¹, Boxin Wang², Jinfeng Li¹, Xiangyu Liu¹, Zhouhan Lin∗³

¹Alibaba Group, China
²University of Illinois at Urbana-Champaign, US
³Shanghai Jiao Tong University, China

{shining.shi, luyang ww, jinfengli jf, eason.lxy}@alibaba-inc.com, boxinw2@illinois.edu,
lin.zhouhan@gmail.com

Abstract

Punctuation restoration is an important post-processing step in automatic speech recognition. Among other kinds of external information, part-of-speech (POS) tags provide informative tags, suggesting each input token’s syntactic role, which has been shown to be beneficial for the punctuation restoration task. In this work, we incorporate an external POS tagger and fuse its predicted labels into the existing language model to provide syntactic information. Besides, we propose sequence boundary sampling (SBS) to learn punctuation positions more efficiently as a sequence tagging task. Experimental results show that our methods can consistently obtain performance gains and achieve a new state-of-the-art on the common IWSLT benchmark. Further ablation studies illustrate that both large pre-trained language models and the external POS tagger take essential parts to improve the model’s performance.

Index Terms: part-of-speech, punctuation restoration, speech recognition

1. Introduction

Punctuation restoration is one of the many post-processing steps in automatic speech recognition (ASR) that are non-trivial to be dealt with. At the meantime, it plays a vital role in improving the readability of the original ASR predicted speech transcripts. Huge efforts have been devoted to investigate better model structures to recover punctuation from raw lexical ASR output, including multi-layer perceptron (MLP) [1], conditional random field (CRF) [2], recurrent neural networks (RNNs) [3, 4, 5, 6, 7, 18, 19, 20], convolutional neural networks (CNNs) [8, 9], and transformers [8, 9, 10]. In addition, a wide range of correlated tasks can be utilized to improve the performance of punctuation restoration via multi-task learning, such as sentence boundary detection [11], capitalization recovering [5], disfluency removing [2], and dependency parsing [12].

Pre-trained language models (LMs) play an increasingly important role in this task. It has been proposed to use Bidirectional Encoder Representations from Transformers (BERT) [13] as a building block and treat punctuation restoration as a sequence tagging task [14], which significantly improved its performance. A series of follow-up works have revealed the effectiveness of other kinds of pre-trained LMs for this task [15, 16].

Although it is possible to recover the punctuation merely from lexical data [7, 17], there is other information external to it that we can utilize. Apart from the aforementioned multi-task learning schemes, multi-modality is another practical approach to fuse relevant information from different modalities, which in return leads to improved performance. By multimodal learning, prosodic cues have been proven informative in improving the quality of punctuation restoration [3, 18, 19, 20]. However, although bunch of works have been proposed in multimodal learning with both lexical and acoustic features, fusing part-of-speech (POS) tag knowledge into raw lexical ASR output has not been well studied yet. As one of the important information that reveals the lexical roles of each word in a sequence, POS tags are believed to be beneficial for this task. For instance, typically a sentence will not end with a definite article like “a”, “an”, or “the”. Previous research explored using POS tag prediction as an auxiliary task in a multi-task learning scenario, and have found that such multi-task learning works for both of tasks [21]. Instead, we propose to explicitly use an external POS tagger to enhance textual input for punctuation restoration, enabling the model to incorporate POS tags information while learning through a more straightforward single task learning scenario.

In this work, we present a framework to involve POS tags provided by an external POS tagger as an extra input and combine it with a new pre-trained LM, Funnel Transformer [22], which effectively filters out the sequential redundancy. Our contributions are as follows:

- We propose a novel framework to employ an external POS tagger to provide syntactic information for punctuation restoration, as well as a new stochastic sampling scheme called sequence boundary sampling (SBS) to better adapt to pre-trained LMs. With RoBERTa [23], our method sets a new state-of-the-art on IWSLT datasets in terms of Micro F₁.
- We introduce Funnel Transformer [22] to our framework and further push the gap between our method and previous studies.
- As ablation study, we examine the punctuation restoration performance of a wide range of pre-trained LMs in a fair and comparable setting, which provides a wide set of pre-trained LM benchmarks on this task.

2. Method

Whether a word needs to be followed by punctuation is closely related to its grammatical role. For instance, a comma is often placed before the coordinating conjunction to join two independent clauses. In this section, we introduce how we incorporate an external POS tagger for punctuation restoration. Specifically, our framework consists of a POS fusion module and the SBS batch sampling strategy.

**We've made the source code and detailed evaluation results of this work publicly available at [https://github.com/ShiningLab/POS-Tagger-for-Punctuation-Restoration.git]**
Our model forms punctuation restoration as a sequence tagging task, which incorporates both a pre-trained LM and a trained POS tagger for final punctuation tag prediction. Assume that we have a sequence of input \( X \) of length \( n \), a pre-trained LM with hidden size \( d \), and a neural POS tagger with hidden size \( b \). Figure 1 is a visual illustration of our model.

2.1. Fusing POS tags into LM representations

On the LM side, we make use of the LM hidden states. Thus we can view the LM as a function \( F \) parameterized by \( \theta \), mapping the sequence \( X \) into a sequence of context dependent embeddings \( H \) by

\[
H = F_\theta(X) \in \mathbb{R}^{n \times d},
\]

where \( H \) is the last layer hidden states of the given LM. In subsequent steps, we will combine \( H \) with the information from the POS tagger side.

On the POS tagger side, we leverage a neural POS tagger by using its predictions as well as its softmax layer weights. Formally, the POS tag predictions \( \hat{T} \) are produced by

\[
\hat{T} = F_W(X) \in \mathbb{R}^n,
\]

where \( F_W(\cdot) \) stands for the POS tagger, with \( W \in \mathbb{R}^{b \times b} \) being its softmax layer weights. This weight matrix can be viewed as a POS embedding matrix, with each embedding having a size \( b \) and \( e \) being the number of POS tag classes.

Further, in the fusion step, we first concatenate \( H \) and \( E \) alongside the sequential dimension to get a fused representation \( C \in \mathbb{R}^{n \times (d+b)} \). Different from conventional practices that combine high-level representations by concatenation alone \([5, 9]\), which may suffer from the inefficacy to model the cross-modality relationship, we utilize a self-attention encoder \([8]\) as the fusion layer that enables both features to better interact with each other through the multi-head self-attention mechanism.

2.2. Sequence boundary sampling

Since sentence boundaries are not explicit in raw ASR output, the raw output of the whole training set can be viewed as a continuous word stream. Due to memory constraints, it have to be truncated to align with a maximum sequence length \( L \). Some previous works split the corpus into multiple sequences in pre-processing steps \([14, 16]\), resulting in reusing the same truncation of the training set for every epoch. Some others rotates the training set one token at a time between different epochs \([13]\), however yielding the training sample size too large. To further randomize the truncation of the continuous word stream, we propose SBS, where we uniformly select a range in the corpus \( S \), starting from \( x_{k} \in [i| L] \) to \( x_{k}+L-1 \), forming a token sequence \( X = \{x_{j},...,x_{j+L-1}\} \) of length \( L \). We limit the number of sampling times to \( 1 \) so as to maintain an acceptable training size while keeping the possibility of exposing every token at every sequence position to the model for more robust learning.

This sampling mechanism provides a computationally more efficient process than earlier ways by both weakening the connection between positions and tokens and allowing mini-batches of samples to represent the entire corpus.

3. Experiments

In this section, we conduct experiments on IWSLT dataset, as well as ablation studies to investigate the efficacy of SBS, POS fusion and several pre-trained LMs respectively. Experimental results demonstrate that our proposed methods with SBS and POS fusion can achieve state-of-the-art performance on IWSLT datasets. Finally, we conduct case studies to further evaluate some succeed and failure cases.

3.1. Experimental setup

Dataset. We evaluate our methods on transcriptions of TED Talks from IWSLT datasets \([25]\), which has been regarded as
Table 1: An example of pre-processed data to align with BERT (bert-base-uncased).

| Raw Word Sequence | Raw Label Sequence | Token Sequence (s) | Label Sequence (%) | POS Tag Sequence (%) |
|-------------------|-------------------|-------------------|-------------------|-------------------|
| texting            | O O O O            | (BOS) o o o o     | 0 0 0 0           | O O O O           |
|               | PRF               | PERIOD            | 0 0 0 0           | O O O O           |
|               | PRF               | PERIOD            | 0 0 0 0           | O O O O           |
|               | PRF               | PERIOD            | 0 0 0 0           | O O O O           |

Table 2: Evaluation results on Ref. in terms of P(%), R(%), Micro F1(%), and Mean F1(%).

| Language Model | Modification | COMMA | PERIOD | QUESTION | Overall |
|---------------|--------------|-------|--------|----------|---------|
|                |              | P    | R     | F      | P      | R     | F      | P      | R     | F      | P      | R     | F      | Overall |
| None           |              |      |       |        |        |      |       |        |      |       |        |      |       |        |         |
|                |              |      |       |        |        |      |       |        |      |       |        |      |       |        |         |
|                |              |      |       |        |        |      |       |        |      |       |        |      |       |        |         |
|                |              |      |       |        |        |      |       |        |      |       |        |      |       |        |         |

1We download data from https://github.com/xashru/punctuation-restoration

2https://huggingface.co/flair/upos-english-fast

3We consider reported F1-score as Micro F1 if it does not match Mean F1.

4We consider reported F1-score as Micro F1 if it does not match Mean F1.

5We consider reported F1-score as Micro F1 if it does not match Mean F1.

6We consider reported F1-score as Micro F1 if it does not match Mean F1.

7We consider reported F1-score as Micro F1 if it does not match Mean F1.

8We consider reported F1-score as Micro F1 if it does not match Mean F1.

9We consider reported F1-score as Micro F1 if it does not match Mean F1.

10We consider reported F1-score as Micro F1 if it does not match Mean F1.

11We consider reported F1-score as Micro F1 if it does not match Mean F1.

12We consider reported F1-score as Micro F1 if it does not match Mean F1.

13We consider reported F1-score as Micro F1 if it does not match Mean F1.

14We consider reported F1-score as Micro F1 if it does not match Mean F1.

15We consider reported F1-score as Micro F1 if it does not match Mean F1.

16We consider reported F1-score as Micro F1 if it does not match Mean F1.

17We consider reported F1-score as Micro F1 if it does not match Mean F1.

18We consider reported F1-score as Micro F1 if it does not match Mean F1.

19We consider reported F1-score as Micro F1 if it does not match Mean F1.

20We consider reported F1-score as Micro F1 if it does not match Mean F1.

21We consider reported F1-score as Micro F1 if it does not match Mean F1.

22We consider reported F1-score as Micro F1 if it does not match Mean F1.

23We consider reported F1-score as Micro F1 if it does not match Mean F1.

24We consider reported F1-score as Micro F1 if it does not match Mean F1.

25We consider reported F1-score as Micro F1 if it does not match Mean F1.

26We consider reported F1-score as Micro F1 if it does not match Mean F1.

27We consider reported F1-score as Micro F1 if it does not match Mean F1.

28We consider reported F1-score as Micro F1 if it does not match Mean F1.

29We consider reported F1-score as Micro F1 if it does not match Mean F1.

30We consider reported F1-score as Micro F1 if it does not match Mean F1.

31We consider reported F1-score as Micro F1 if it does not match Mean F1.

32We consider reported F1-score as Micro F1 if it does not match Mean F1.

33We consider reported F1-score as Micro F1 if it does not match Mean F1.

34We consider reported F1-score as Micro F1 if it does not match Mean F1.

35We consider reported F1-score as Micro F1 if it does not match Mean F1.

36We consider reported F1-score as Micro F1 if it does not match Mean F1.

37We consider reported F1-score as Micro F1 if it does not match Mean F1.

38We consider reported F1-score as Micro F1 if it does not match Mean F1.

39We consider reported F1-score as Micro F1 if it does not match Mean F1.

40We consider reported F1-score as Micro F1 if it does not match Mean F1.

41We consider reported F1-score as Micro F1 if it does not match Mean F1.

42We consider reported F1-score as Micro F1 if it does not match Mean F1.

43We consider reported F1-score as Micro F1 if it does not match Mean F1.

44We consider reported F1-score as Micro F1 if it does not match Mean F1.

45We consider reported F1-score as Micro F1 if it does not match Mean F1.

46We consider reported F1-score as Micro F1 if it does not match Mean F1.

47We consider reported F1-score as Micro F1 if it does not match Mean F1.
method, i.e., with POS tag fusion and SBS, and compare that to the corresponding published baselines using the same pre-trained LM. Our models are denoted as Ours: POS fusion + SBS in each of the groups.

The last group of each table is named funnel-transformer-xlarge. In this group, we provide our method using Funnel Transformer as its LM. Since there are no preceding baseline lines to compare, it is a good testbed setting for our method. We conduct four ablation studies with Funnel Transformer to show the relative contributions of each component in our method. We denote the base setting, which consists of only an LM and a linear layer, as None, and SBS for this base setting with SBS during training. We refer to -POS embedding + SBS as our model involving POS tags as part of inputs but with POS embedding initialized with random noise. The final line denoted by POS Fusion + SBS is the full setting of our method.

For punctuation restoration on the reference test set, we report the evaluation results in Table 2. Firstly, our method in the group of RoBERTa (roberta-large) outperforms all the previous models in terms of the overall Micro $F_1$ and on-par with them in terms of Mean $F_1$. Further incorporating Funnel Transformer, our model achieves the new state-of-the-art, resulting in an absolute improvement of 1.4% in both Micro $F_1$ and Mean $F_1$ compared to previous best [16]. Note that this is without focal loss and data augmentation, which suggests that it can be further pushed up. Compared with [16], our method consistently improves performances in all three individual classes.

For punctuation restoration on the ASR test set, we report the evaluation results in Table 3. Without using data augmentation, our funnel-transformer-xlarge based model obtains 72.6% in Micro $F_1$ and 70.9% in Mean $F_1$, outperforming previous best roberta-large based model [16] by absolute 1.3% and 3.6% separately. In terms of Mean $F_1$, our final version with Funnel Transformer achieves competitive results, the only previous one that outperforms our method is through adversarial learning [21]. Moreover, the aforementioned trend on Ref. can also be noticed on ASR as well, including the positive impact of large LMs, SBS, and POS Fusion with POS tagger provided POS embeddings. Compared to the ref test set, the primary extra difficulty in ASR test set is the noise caused by the incorrectly predicted words. Since POS is an attribute of the natural language, our method thus exhibits heavy dependency on the preceding ASR outputs. An incorrect word will likely be assigned with an incorrect POS tag, leading to a misrepresentation of both lexical and POS features. It can be seen that data augmentation techniques that simulate error words created by the ASR system play an essential role in handling the noiser test set. We believe our model can also benefit from data augmentation, but we leave it for future work.

As for ablation studies, we analyze different settings within the Funnel Transformer group. Firstly, the base setting, which is Funnel Transformer simply followed by a liner layer, is already performing satisfyingly well, especially for COMMA. With SBS, the $F_1$ score for QUESTION goes up from 86.6% to 90.7% without too much loss on the other two classes. This indicates that SBS does help to alleviate the class imbalance issue. Particularly in training samples generated by SBS, PERIOD and QUESTION no longer frequently appear near the end of the sequence. Thus models have to focus on contexts rather than positions to infer punctuations. In contrast to -POS embedding + SBS equipped with a random initialized POS embedding, our full setting POS Fusion + SBS performs the best across all settings. One possible reason is that pre-trained weights, instead of random weights, can serve as regularization or constraints for models to focus on major features from training samples and accelerate training before overfitting. We want to stress that it is stated in former studies [17][18] that base LMs are preferred due to the small size of IWSLT datasets. However, our findings suggest the opposite.

4. Conclusion

We propose a novel framework that brings POS knowledge via a self-attention based fusion layer for punctuation restoration. Experiments conducted on IWSLT datasets prove that incorporating POS tags makes it possible for prior lexical-based approaches to earn significant performance gains. We also introduce a new sampling technique, SBS, that makes fuller use of the corpus and better adapts to LMs. Empirical results show that our method with Funnel Transformer is superior in performance to all former published works.

5. Acknowledgements

We give thanks to Yichen Gong, Dawei Wang and Rong Zhang for sharing their pearls of wisdom. This work was supported by Shining Lab and Alibaba Group through Alibaba Research Intern Program.

### Table 3: Evaluation results on ASR in terms of $P(\%)$, $R(\%)$, Micro $F_1(\%)$, and Mean $F_1(\%)$

| Language Model | Modification                           | COMMA | PERIOD | QUESTION | Overall |
|---------------|---------------------------------------|-------|--------|----------|---------|
|               |                                       | $P$   | $R$    | $F_1$    | $P$    | $R$    | $F_1$    | $P$ | $R$    | $F_1$    | $P$ | $R$    | $F_1$    |
| None          |                                       | 59.6  | 42.9   | 49.9     | 70.7   | 72.0   | 71.3     | 60.7 | 58.4   | 56.1     | 66.0 | 57.3   | 61.4     |
| Teacher-Ensemble [21] |                       | 60.6  | 58.3   | 59.4     | 71.7   | 72.9   | 72.3     | 66.2 | 55.8   | 60.6     | 66.2 | 62.3   | 64.1     |
| Self-attention [2] |                             | 64.0  | 59.6   | 60.3     | 75.5   | 75.6   | 75.6     | 72.6 | 65.9   | 69.1     | 70.7 | 67.1   | 68.8     |
| bert-base-uncased | Adversarial [21]                  | 70.7  | 68.1   | 69.4     | 77.6   | 77.5   | 77.5     | 68.4 | 66.0   | 67.2     | 72.2 | 70.5   | 71.4     |
|               | FL [17]                            | 59.0  | 76.6   | 66.7     | 78.7   | 79.9   | 79.3     | 60.5 | 71.5   | 65.6     | 66.1 | 76.0   | 70.7     |
|               | Bi-LSTM [25]                      | 49.3  | 62.4   | 55.8     | 75.3   | 76.3   | 75.8     | 44.7 | 60.0   | 51.2     | 60.4 | 70.0   | 64.9     |
|               | Ours: POS Fusion + SBS            | 49.3  | 65.6   | 56.3     | 73.6   | 78.8   | 76.1     | 48.9 | 62.8   | 59.0     | 61.0 | 73.0   | 65.4     |
| bert-large-uncased | Bi-LSTM [25]                   | 49.9  | 67.0   | 57.2     | 77.0   | 78.9   | 77.9     | 50.0 | 74.3   | 59.8     | 61.4 | 73.0   | 66.7     |
|               | Ours: POS Fusion + SBS            | 54.7  | 64.3   | 59.1     | 75.8   | 82.5   | 79.0     | 48.8 | 60.0   | 53.9     | 64.6 | 73.2   | 68.4     |
| roberta-base | Bi-LSTM [25]                      | 51.9  | 69.3   | 59.3     | 77.5   | 80.3   | 78.9     | 50.0 | 65.7   | 56.8     | 62.8 | 74.7   | 68.2     |
|               | Ours: POS Fusion + SBS            | 55.5  | 68.7   | 61.4     | 78.0   | 81.1   | 79.5     | 51.1 | 68.6   | 58.5     | 65.5 | 74.8   | 69.8     |
| roberta-large | Bi-LSTM + augmentation            | 56.6  | 67.9   | 61.8     | 78.7   | 83.1   | 81.9     | 46.6 | 77.1   | 58.1     | 66.5 | 76.7   | 71.3     |
|               | Ours: POS Fusion + SBS            | 56.1  | 68.8   | 66.3     | 81.0   | 83.7   | 82.3     | 55.3 | 74.3   | 63.4     | 72.0 | 76.2   | 74.0     |
| funnel-transformer-xlarge | None                          | 52.6  | 76.5   | 62.3     | 81.2   | 81.8   | 81.5     | 53.1 | 74.3   | 61.9     | 64.1 | 79.1   | 70.8     |
|               | SBS                                | 54.4  | 72.8   | 62.3     | 81.0   | 82.9   | 82.0     | 59.6 | 80.0   | 68.3     | 65.9 | 77.9   | 71.4     |
|               | POS embedding + SBS               | 54.8  | 73.4   | 62.8     | 80.7   | 85.3   | 82.9     | 54.7 | 82.9   | 65.9     | 66.0 | 79.5   | 72.1     |
|               | POS Fusion + SBS                  | 56.6  | 71.6   | 63.2     | 79.0   | 87.8   | 82.8     | 60.5 | 74.3   | 66.7     | 66.9 | 79.3   | 72.6     |

* denotes $p-value < 0.05$ with respect to the corresponding published baseline.
6. References

[1] X. Che, C. Wang, H. Yang, and C. Meinel, “Punctuation prediction for unsegmented transcript based on word vector,” in Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16), 2016, pp. 654–658.

[2] E. Cho, K. Kilgour, J. Niehues, and A. Waibel, “Combination of nn and crf models for joint detection of punctuation and disfluencies,” in Sixteenth annual conference of the international speech communication association, 2015.

[3] O. Tilk and T. Alumae, “Lstm for punctuation restoration in speech transcripts,” in Sixteenth annual conference of the international speech communication association, 2015.

[4] O. Tilk and T. Alumae, “Bidirectional recurrent neural network with attention mechanism for punctuation restoration,” in Interspeech 2016, 2016, pp. 3047–3051. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-1517

[5] V. Pahua, A. Laha, S. Mirkin, V. Raykar, L. Kotlerman, and G. Lev, “Joint learning of correlated sequence labeling tasks using bidirectional recurrent neural networks,” in Proc. Interspeech 2017, 2017, pp. 548–552.

[6] F. Wang, W. Chen, Z. Yang, and B. Xu, “Self-attention based network for punctuation restoration,” in 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, 2018, pp. 2803–2808.

[7] S. Kim, “Deep recurrent neural networks with layer-wise multi-head attentions for punctuation restoration,” in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 7280–7284.

[8] A. Vasswani, 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 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008.

[9] J. Yi and J. Tao, “Self-attention based model for punctuation prediction using word and speech embeddings,” in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 7270–7274.

[10] 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.

[11] 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.

[12] D. Zhang, S. Wu, N. Yang, and M. Li, “Punctuation prediction with transition-based parsing,” in Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2013, pp. 752–760.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[14] K. Maklaja, 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.

[15] 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. Online: Association for Computational Linguistics, Jul. 2020, pp. 272–279.

[16] T. Alam, A. Khan, and F. Alam, “Punctuation restoration using transformer models for high- and low-resource languages,” in Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020). Online: Association for Computational Linguistics, Nov. 2020, pp. 132–142.

[17] J. Yi, J. Tao, Z. Tian, Y. Bai, and C. Fan, “Focal loss for punctuation prediction,” in Proc. Interspeech 2020, pp. 721–725, 2020.

[18] T. Levy, V. Silber-Varod, and A. Moyal, “The effect of pitch, intensity and pause duration in punctuation detection,” in 2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel. IEEE, 2012, pp. 1–4.

[19] G. Szaszak 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] M. Sunkara, S. Ronanoki, D. Bekal, S. Bodapati, and K. Kirchhoff, “Multimodal semi-supervised learning framework for punctuation prediction in conversational speech,” in Proc. Interspeech 2020, 2020.

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

[22] Z. Dai, G. Lai, Y. Yang, and Q. Le, “Funnel-transformer: Filtering out sequential redundancy for efficient language processing,” in Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 4271–4282.

[23] 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.

[24] J. Yi, J. Tao, Z. Wen, Y. Li et al., “Distilling knowledge from an ensemble of models for punctuation prediction,” in Interspeech, 2017, pp. 2779–2783.

[25] M. Federico, M. Cettolo, L. Bentivogli, P. Michael, and S. Sebastian, “Overview of the iwslt 2012 evaluation campaign,” in IWSLT-International Workshop on Spoken Language Translation, 2012, pp. 12–33.

[26] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtovecz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jeremie, J. Phu, C. Xu, T. L. Scoa, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, “Transformers: State-of-the-art natural language processing,” in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45.

[27] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “Albert: A lite bert for self-supervised learning of language representations,” ICLR, 2020.

[28] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, P. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, “Unsupervised cross-lingual representation learning at scale,” in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, Jul. 2020, pp. 8440–8451.

[29] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, and R. Vollgraf, “Flair: An easy-to-use framework for state-of-the-art nlp,” in NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), 2019, pp. 54–59.

[30] L. Prechelt, “Early stopping-but when?” in Neural Networks: Tricks of the trade. Springer, 1998, pp. 55–69.

[31] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, Y. Bengio and Y. LeCun, Eds., 2015.

[32] R. Pascu, T. Mikelov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ser. ICML’13. JMLR.org, 2013, p. III–1310–1318.