Incorporating External POS Tagger for Punctuation Restoration

Ning Shi¹, Wei Wang¹, Boxin Wang², Jinfeng Li³, Xiangyu Liu¹, Zhouhan Lin³*

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

{shining.shi, luyang ww, jinfengli jlf, eason lixy}@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) taggers 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], convolutional neural networks (CNNs) [1, 6], and transformers [8, 6, 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 work¹, 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.

¹We’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.

Corresponding Author.
where $F_W(\cdot)$ stands for the POS tagger, with $W \in \mathbb{R}^{n \times b}$ 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. For simplicity, we only show the related parameters $W$ in the POS tagger. To get POS embeddings $E \in \mathbb{R}^{n \times b}$ for the input sequence, we use elements in $T$ to lookup for the corresponding columns in $W$, and form the POS embedding $E \in \mathbb{R}^{n \times b}$.

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 [3, 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. Subsequently, we pass $C$ through the block of self-attention layer $L_\gamma$, as well as a final softmax layer $S_\eta$ to output punctuation tags $\hat{Y}$ by

$$\hat{Y} = S_\eta(L_\gamma(C)),$$

where $\gamma$ and $\eta$ stand for the corresponding parameters in the components. Training is simply conducted as back-propagating the cross entropy loss between $\hat{Y}$ and its corresponding ground truth $Y$, over parameter $\theta, W, \gamma$, and $\eta$, where $\theta$ is initialized from the pre-trained LM, and $W$ is initialized from the external POS tagger weights.

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 has 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 [15], 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 $x_{0 \leq j < |S| - L}$ to $x_{j + L - 1}$, forming a token sequence $X = \{x_0, \ldots, x_{j + L - 1}\}$ of length $L$. We limit the number of sampling times to $|S| - L$ 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).

<table>
<thead>
<tr>
<th>Raw Word Sequence</th>
<th>Raw Label Sequence</th>
<th>Token Sequence (S)</th>
<th>Label Sequence (Y)</th>
<th>POS Tag Sequence (T)</th>
<th>Position Mark</th>
</tr>
</thead>
<tbody>
<tr>
<td>[CLS] x [SEP]</td>
<td>[CLS] x [SEP]</td>
<td>[CLS] x [SEP]</td>
<td>[CLS] x [SEP]</td>
<td>[CLS] x [SEP]</td>
<td>[CLS] x [SEP]</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Language Model</th>
<th>Modification</th>
<th>COMMA</th>
<th>PERIOD</th>
<th>QUESTION</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bert-base-uncased</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bert-large-uncased</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>roberta-base</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>roberta-large</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>funnel-nerformer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

1. A standard benchmark on punctuation restoration. Following the standard evaluation protocol, we use the same splits: 2.1M words for training, 296K words for validation, 12626 words for the manual transcription test set (Ref.), and 12822 words for the actual ASR transcription test set (ASR) provided in [1], where the wrong predicted words of ASR cause the task more challenging. There are four labels for each word, meaning which punctuation to follow behind. Specifically, (i) COMMA is for commas, colons, and dashes; (ii) PERIOD is for full stops, exclamation marks, and semicolons; (iii) QUESTION is for question marks only; (iv) O is for no punctuation.

2. Preprocessing. To maintain an original-to-tokenized alignment with the subword tokenization of LMs, we assign label O for all non-tail subword pieces in a word and keep the original punctuation label only for the tail subword piece. Moreover, every tokenized sentence is also prefixed with a beginning-of-sentence token [bos] and suffixed with an end-of-sentence token [eos]. Thanks to SBS, it is unnecessary to hold a pad token since token sequences are always of the maximum sequence length. We maintain a position mask to filter out non-tail subword pieces and special tokens from evaluation. A pre-processed data example after aligning with BERT is shown in Table 1.

3. Evaluation Metrics. We measure the performance in terms of precision (P), recall (R), and F1-score. We notice the definition of the F1-score is not consistent in previous studies, thus, to be on the same page, we report both micro F1-score (Micro F1) and mean F1-score (Mean F1). The latter is defined as the average F1 score of COMMA, PERIOD and QUESTION. The Mean F1 is manually calculated for preceding approaches that did not state the definition of reported F1-score. With the exception of those clearly noted, we consider reported F1-score as Micro F1 if it does not match Mean F1 we calculated. Otherwise, we level it as Mean F1 in our evaluation table.

4. Implementation Details. We use Transformers [26] built by HuggingFace to tokenize raw words into subword units and explore different LMs involving BERT [13], ALBERT [27], RobBERTa [23], XLM-RoBERTa [28], and Funnel Transformer [22]. In our experiment, we use the fast universal POS tagging model for English by Flair [29], where b = 512 and e = 20. For our self-attention based fusion layer, in base LMs (d < 1024), the dimension of the feedforward network and the number of attention heads are 3072 and R respectively. Accordingly, in large LMs (d ≥ 1024), they are set to 4096 and 16, respectively. The residual dropout rate is 0.1. We train on a single NVIDIA Tesla V100 with a maximum sequence length of 256 and a batch size of 8. Batches containing less than 8 samples are dropped. To avoid overfitting, we adopt early stopping [30] with a patience of 4 epochs for a lower validation loss. We use Adam optimizer [31] with a learning rate of 5e−6 and an ℓ2 gradient clipping of 5.0 [32]. We choose to minimize a cross entropy loss instead of focal loss [17] for a fair comparison with previous results that do not make use of focal loss.

3.2. Results

Our results on reference text and on ASR outputs are shown in Table 2 and 3, respectively. Since pre-trained LMs play a vital role in the performances, reported results are grouped according to their LMs. In each group, we list the complete version of our

---

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

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 with Funnel Transformer [22] as its LM. Since there are no preceding baselines 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 F1, and on-par with them in terms of Mean F1. Further incorporating Funnel Transformer, our model achieves the new state-of-the-art, resulting in an absolute improvement of 1.4% in both Micro F1, and Mean F1, 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 F1, and 70.9% in Mean F1, outperforming previous best roberta-large based model [16] by absolute 1.3% and 3.6% separately. In terms of Mean F1, 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 model thus exhibits heavier 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 noisier 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 linear layer, is already ready performing satisfyingly well, especially for COMMA. With SBS, the F1 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, 15] 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.
6. References


