

Improving Punctuation Restoration for Speech Transcripts via External Data

Xue-Yong Fu, Cheng Chen, Md Tahmid Rahman Laskar,
Shashi Bhushan TN, Simon Corston-Oliver

Dialpad Canada Inc.

Vancouver, BC, Canada

{xue-yong, cchen, tahmid.rahman}@dialpad.com

{sbhushan, scorston-oliver}@dialpad.com

Abstract

Automatic Speech Recognition (ASR) systems generally do not produce punctuated transcripts. To make transcripts more readable and follow the expected input format for downstream language models, it is necessary to add punctuation marks. In this paper, we tackle the punctuation restoration problem specifically for the noisy text (e.g., phone conversation scenarios). To leverage the available written text datasets, we introduce a **data sampling technique based on an n-gram language model** to sample more training data that are similar to our in-domain data. Moreover, we propose a **two-stage fine-tuning** approach that utilizes the sampled external data as well as our in-domain dataset for models based on BERT. Extensive experiments show that the proposed approach outperforms the baseline with an improvement of 1.12% F1 score.

1 Introduction

ASR systems are widely deployed in many products to support a variety of business applications, such as customer support, virtual assistants, and conversational systems. The context for this research is to build commercial products that perform realtime transcription of English business telephone calls, with additional natural language processing to enable features such as assistance for customer support agents and note-taking during business meetings. Though ASR systems trained on large amounts of data can transcribe human speech with high accuracy (Kannan et al., 2019; Baevski et al., 2020), spoken dialogue transcripts produced by such systems often exclude punctuation marks. Therefore, the user reading a long unpunctuated transcript may find it unreadable (Zelasko et al., 2018; Alam et al., 2020).

Beyond the need for readability, downstream natural language understanding (NLU) models use transcripts to provide more business insights. Many NLU models (e.g., sentiment analysis and named

entity recognition (NER)) achieving state-of-the-art performance are based on large-scale language models (e.g. BERT (Devlin et al., 2019)), which are often pre-trained on well-punctuated texts such as English Wikipedia and BookCorpus (Zhu et al., 2015). However, NLU models trained on such datasets may not perform well on unpunctuated transcripts. Thus, for the conversational domain, punctuated speech transcripts are also necessary.

While most prior research has focused on automatically punctuating monologue speech transcripts, our work is focused on **transcripts of real-world phone conversations with turn-taking between two or more people**. Spontaneous turn-taking conversations typically are more noisy than monologues because they might contain speech overlaps, false starts, repetitions, and higher transcription word error rates. One way to address this problem is to obtain large amounts of human-punctuated transcripts as training data. However, human annotation is expensive and time-consuming. In this paper, we propose a novel way to leverage freely available external data to improve the punctuation restoration accuracy in human-to-human business conversational data. The main contributions of this paper are as follows:

- We introduce a **new data sampling technique** based on an n-gram language model to select additional training data relevant to our target domain from publicly available datasets. Then we propose a two-stage fine-tuning approach based on BERT to train our model.
- Extensive experiments show that the proposed method is effective to improve performance on unpunctuated noisy data collected from an ASR system. Since our goal is to develop a punctuation restoration model for real-time scenarios in production, we also **analyze the performance of different layers in BERT** and observe that layer reduction significantly im-

proves the inference speed while maintaining the accuracy. As a secondary contribution, we demonstrate the production environment that we use to deploy the trained model.

2 Related Work

In this section, we first review the recent work on punctuation restoration. Since we sample additional training data from external corpora, we also review related work on data sampling.

2.1 Punctuation Restoration

Most of the punctuation restoration models formulate the problem as a sequence labeling task (Hentschel et al., 2021; Yi et al., 2020; Chen et al., 2021b; Huang et al., 2021; Nagy et al., 2021). The model architectures vary from the RNNs to the pre-trained contextualized language models. In recent years, Tilk and Alumäe (2015), Salimbajevs (2016), and Xu et al. (2016) used long short-term memory units to restore punctuation in speech transcripts, while Courtland et al. (2020) and Cai and Wang (2019) used a bidirectional transformer (Vaswani et al., 2017) to predict punctuation marks.

2.2 External Training Data Sampling

For training data sampling, Chiu et al. (2016) and Karimi et al. (2017) used the data that were similar to the biology domain. To measure similarity between two datasets, Dai et al. (2019) studied different techniques, such as: vocabulary overlaps, language model perplexity, and word vector variance. They also investigated how similarity measures can be used to sample in-domain pre-training data. Moreover, for NER, they observed that the second phase of in-domain pre-training led to performance gains under both high and low-resource settings. Recently, Han and Eisenstein (2019) proposed domain-adaptive fine-tuning in which contextualized embeddings were adapted by masked language modeling on texts from the target domain.

3 Method

We frame punctuation restoration as a sequence labeling problem. There are four classes that the model needs to predict, including **PERIOD**¹, **COMMA**, **QUESTIONMARK**, and **NONE**. More specifically, the model predicts what the punctuation mark next to a given token is. If

¹Although there are few exclamation marks in the data, we convert them to periods for simplicity.

there is no punctuation mark after the input token, the prediction should be **NONE**.

In Figure 1, we show the overall architecture of our proposed model that utilizes a two stage fine-tuning approach: First, (i) we do data sampling from an external dataset (e.g., movie subtitles) using our proposed n-gram similarity approach and fine-tune a pre-trained BERT model. Then, (ii) we annotate unpunctuated phone conversation transcripts generated from an ASR system to construct an in-domain training set (noisy data) to apply the second stage of fine-tuning using the BERT model fine-tuned in the previous stage. Below, we describe the data sampling, data annotation, and two-stage fine-tuning approach in detail.

3.1 Data Sampling from the External Dataset via N-gram Language Model

In principle, we can obtain a tremendous amount of punctuated text from the internet (e.g English Wikipedia). However, written English is fundamentally different from spoken English and training a model on a corpus that is distant from the target domain might end up harming model performance. Thus, we choose to use a movie subtitle corpus, OpenSubtitles (Lison and Tiedemann, 2016), as our first stage fine-tuning data because there are many human-to-human conversations in movies.

As movie conversations are not necessarily business conversations, training a model on out-of-domain data might also degrade model accuracy. Thus, we propose a data sampling method based on language model perplexity to select data similar to our target domain (noisy phone conversations). Some examples from the out-of-domain (movie subtitles) data and the in-domain (noisy phone conversations) data are shown in Table 1.

For data sampling, we follow (Dai et al., 2019) and assign a probability to any sequence of words $\langle w_1, \dots, w_N \rangle$ using the n-gram language model (Jurafsky and Martin, 2020). The probability of a sequence can be obtained by chain rule of probability.

$$p(w_1, w_2, \dots, w_N) = \prod_{i=1}^N p(w_i | w_1^{i-1}) \quad (1)$$

Here, N is the length of the sequence and w_1^{i-1} are all words before word w_i . In our work, we use a 4-gram language model.

To sample data that are similar to our in-domain data, we first train a language model on our tar-

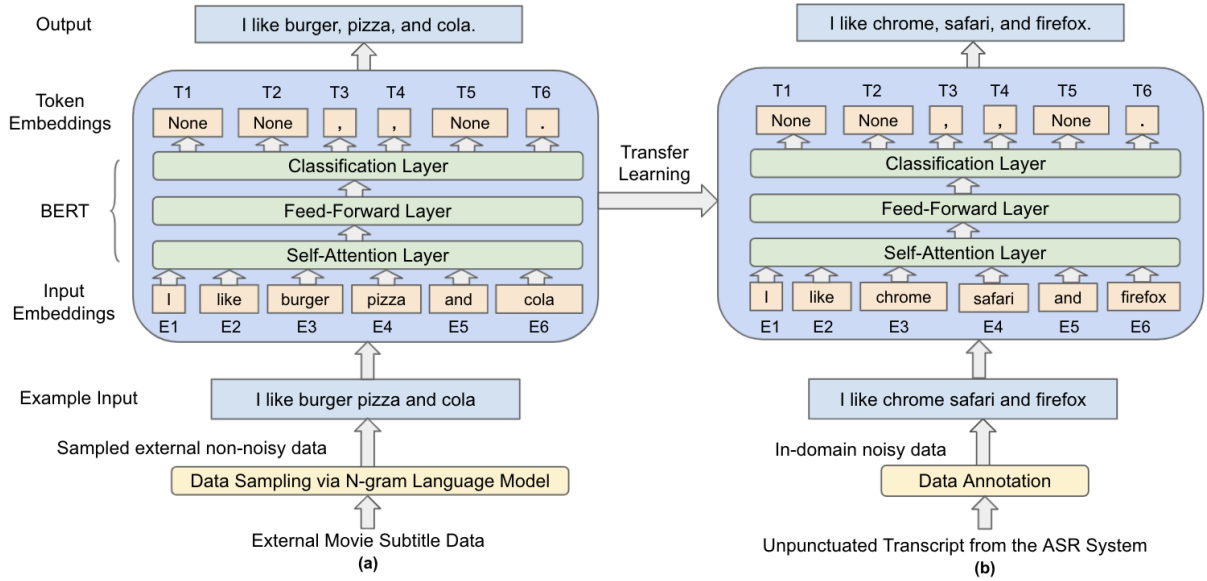


Figure 1: Our two stage fine-tuning approach: (a) at first on the external dataset (e.g., movie subtitles), we sample the external data using n-gram similarity and fine-tune the pre-trained BERT model. (b) Next we annotate unpunctuated transcripts generated from an ASR system to construct the in-domain training set (noisy data) and apply the second stage of fine-tuning using the BERT model fine-tuned in the previous stage.

get domain, which is business phone conversations. Then we evaluate the utterances (subtitles) in the movie subtitle corpus based on the perplexity generated by the model for a given utterance (subtitle) to represent the degree of similarity. The main difference between (Dai et al., 2019) and our work is that we measure the similarity at the utterance level instead of the corpus level. When an utterance is unlikely to occur in the data where the language model is trained on, it will assign a low probability and high perplexity. More specifically, we feed an utterance from the movie subtitle corpus to the 4-gram language model and then the model estimates its perplexity. Afterwards, we rank the subtitles by perplexity and select the top 4.8M subtitles as analogous to in-domain data. Some examples from the external and the in-domain data are shown in Table 1.

3.2 Data Annotation for the ASR Data

We sample 320,000 utterances produced by the ASR model from business phone calls across a time period of a year (see Table 2 for details). We also make sure utterances are distributed homogeneously among different scenarios. More importantly, we follow our standard data privacy protocol while working on data annotation. Since human-to-human conversations are incredibly noisy that could degrade the performance of a pre-trained language model that is trained on written texts, we

Out-of-domain (External Movie Subtitle Dataset)	
Our dog disappeared. Sometimes those serious ones fool you. He looks so utterly vulnerable. God destroyed whole cities to punish man's wickedness. I mean, who comes hereafter dark besides murderers? Your paintings were impressive joker.	
In-domain (Noisy Phone Conversations)	
Well, thank you for coming down. What did you find out so far? I guess you could give me her number. Um, just hold on a second. Might I have your email? Eh, I really don't know what to do about it!	

Table 1: Examples from the out-of-domain and the in-domain datasets.

Data type	Comma	Period	Question
Training	1166386	860403	118203
Dev	15899	14143	1722
Test	6503	3880	629

Table 2: In-domain dataset (noisy data) distribution.

remove repetitions, false starts, and filler words from transcripts to address this problem. Note that inserting punctuation marks to unpunctuated texts might take a long time for an annotator. To speed up this process, we use a punctuation restoration model based on BERT (Devlin et al., 2019) that is trained on a much smaller dataset to produce default punctuation marks for utterances and let the annotators edit these punctuation marks. In total,

Model	First fine-tuning	Second fine-tuning	Precision	Recall	F1 score
Indom	In-domain data	×	75.01	63.71	68.48
Samex	Sampled external data	×	67.70	57.49	61.68
Two-stage	Sampled external data	In-domain data	75.94	65.18	69.60

Table 3: Performance based on different fine-tuning techniques with 6-layer BERT.

102 annotators contributed to the annotation job.

3.3 Two-Stage Fine-tuning Approach

Fine-tuning a pre-trained language model at first in a dataset similar to the target domain, and then fine-tuning it again in the target dataset is found effective in different tasks, such as answer selection (Garg et al., 2020), and query focused text summarization (Laskar et al., 2020a,b). In this paper, we also utilize a two-stage fine-tuning approach that transfers a pre-trained language model to the punctuation restoration task in the first-stage; and then adapts the obtained model to the specific target domain in the second-stage. For that purpose, we use the BERT model (Devlin et al., 2019) that is a Transformer-based (Vaswani et al., 2017) contextual language model as our backbone model. To utilize BERT for sequence labeling, a classification layer is added on top of the last layer (see Figure 1). In addition, as our goal is to deploy a punctuation restoration model in real-time and at scale, we use the weights of the first (bottom) six layers of the BERT base (Devlin et al., 2019) model (12 transformer layers), to initialize a 6-layer BERT model for fine-tuning.

Type	Precision	Recall	F1
Period	72.57	77.89	75.14
Question	84.63	63.91	72.83
Comma	77.23	57.79	66.11
Overall	75.94	65.18	69.60

Table 4: Performance by the Two-stage model on different punctuation marks.

4 Experiments

In our experiments, we investigate the following: (i) Will the model perform better if we use external datasets? (ii) Will the model performance degrade if we reduce the number of layers? Below, we first describe our experimental settings. Then we analyze the model performance when external data are used, followed by discussing the performance when layer reduction is applied.

4.1 Experimental Settings

To fine-tune 6 layer BERT models, we used the following hyperparameters: gradient accumulation steps = 1, warmup proportion = 0.1, learning rate = $5e - 5$, train batch size = 50, and the max sequence length = 300. All training experiments were run in an Google Cloud Platform (GCP) instance using 1 NVIDIA P100 GPU. To investigate the inference speed for real-time scenarios, we ran our experiments with 1 CPU, Intel(R) Xeon(R) CPU@2.20GHz.

4.2 Performance while using External Data

We fine-tune models on various data settings and evaluate them on the test set. In the first data setting, we fine-tune the BERT model only on the in-domain data. We denote this model as Indom. In the second setting, the BERT model, fine-tuned only on the sampled external data is denoted as Samex. The last setting is our proposed Two-stage approach that first fine-tunes BERT on the sampled external data and then fine-tunes again on the in-domain data. The results are shown in Table 3.

The model (Samex) that was trained only on sampled external data unsurprisingly shows the F1 score as 61.68, which is 7.92% lower than the model (Two-stage) that was also fine-tuned on the in-domain data. This indicates that in-domain fine-tuning is crucial to boost model performance in a target domain. We also find that the model (two-stage) fine-tuned on both the in-domain and the sampled external data outperforms the model (Indom) fine-tuned only on the in-domain data by 1.12% F1 score. This demonstrates the effectiveness of two-stage fine-tuning using external data.

Table 4 shows how our best performing model (two-stage) performs in three different punctuation marks with the 6-layer BERT model. It is worth noting that the recall of commas is much lower than other punctuation types. One explanation is that although we tried our best to remove repetitions, false starts, and filler words, there are still a noticeable amount of them in the transcripts. Thus, it often makes it very challenging to place commas.

Model	P	R	F	Inf. time
3-layer	73.49	62.93	67.25	13.83
6-layer	75.01	63.71	68.48	26.52
12-layer	74.27	64.09	68.36	50.01

Table 5: Performance based on different number of BERT layers. Here, Precision, Recall, and F1 are denoted by P, R, and F respectively; while the Inference Time (Inf. time) is based on milliseconds per example.

4.3 Performance based on Layer Reduction

To know the effect of layer reduction in BERT, we fine-tune three BERT models with different numbers of layers (3, 6, 12) on the in-domain data only. This setting is similar to the Indom model except changing the number of layers. The results are shown in Table 5. It is worth noting that there is no significant difference in model performance between the smaller models (3-layer and 6-layer) and the 12-layer model. The 6-layer model even outperforms the 12-layer model in terms of Precision and F1 scores. More importantly, the inference time is significantly decreased by 47.96% and 72.35% from the 12-layer model to the 6-layer and the 3-layer models respectively.

4.4 Production Deployment

For production inference, we also remove the filler words and repetitions similar to the training phase to ensure that the input distributions are similar. For production deployment, we use an Intel(R) Xeon(R) CPU@2.20GHz and feed one utterance to the model at once (i.e., batch size = 1). The model runs in real-time in our production environment. The average inference speed is 201 milliseconds per utterance.

5 Conclusion

In this paper, we propose a new approach to **sample training data that are similar to the target domain** from freely available general corpora via an n-gram language model. We conduct extensive experiments using BERT with different data settings for the punctuation restoration task and find that our two-stage fine-tuning approach is effective to improve model performance. We also show the effectiveness of our model for real-time inference as we find that removing the first 6 layers of the 12 layer BERT does not harm the accuracy but substantially improves the speed. In the future, we will investigate how other external datasets (Chen et al.,

2021a) impact the model performance.

Acknowledgements

We thank Harsh Saini for help with model deployment, Shayna Gardiner for giving ideas about the writing, and Feriba Saboor for help with data annotation.

References

- Tanvirul Alam, Akib Khan, and Firoj Alam. 2020. [Punctuation restoration using transformer models for high-and low-resource languages](#). In *Proceedings of the Sixth Workshop on Noisy User-generated Text, W-NUT@EMNLP 2020 Online, November 19, 2020*, pages 132–142. Association for Computational Linguistics.
- Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. [wav2vec 2.0: A framework for self-supervised learning of speech representations](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Yunqi Cai and Dong Wang. 2019. Question mark prediction by bert. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 363–367. IEEE.
- Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2021a. Summscreen: A dataset for abstractive screenplay summarization. *arXiv preprint arXiv:2104.07091*.
- Qian Chen, Wen Wang, Mengzhe Chen, and Qinglin Zhang. 2021b. Discriminative self-training for punctuation prediction. *arXiv preprint arXiv:2104.10339*.
- Billy Chiu, Gamal K. O. Crichton, Anna Korhonen, and Sampo Pyysalo. 2016. [How to train good word embeddings for biomedical NLP](#). In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing, BioNLP@ACL 2016, Berlin, Germany, August 12, 2016*, pages 166–174. Association for Computational Linguistics.
- Maury Courtland, Adam Faulkner, and Gayle McElvain. 2020. [Efficient automatic punctuation restoration using bidirectional transformers with robust inference](#). In *Proceedings of the 17th International Conference on Spoken Language Translation, IWSLT 2020, Online, July 9 - 10, 2020*, pages 272–279. Association for Computational Linguistics.
- Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cécile Paris. 2019. [Using similarity measures to select pre-training data for NER](#). In *Proceedings of the 2019 Conference of the North American Chapter of the*

- Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1460–1470. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.
- Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. [TANDA: transfer and adapt pre-trained transformer models for answer sentence selection](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7780–7788. AAAI Press.
- Xiaochuang Han and Jacob Eisenstein. 2019. [Unsupervised domain adaptation of contextualized embeddings for sequence labeling](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4237–4247. Association for Computational Linguistics.
- Michael Hentschel, Emiru Tsunoo, and Takao Okuda. 2021. Making punctuation restoration robust and fast with multi-task learning and knowledge distillation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7773–7777. IEEE.
- Qiushi Huang, Tom Ko, H Lilian Tang, Xubo Liu, and Bo Wu. 2021. Token-level supervised contrastive learning for punctuation restoration. *arXiv preprint arXiv:2107.09099*.
- Daniel Jurafsky and James H Martin. 2020. [Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition](#).
- Anjuli Kannan, Arindrima Datta, Tara N. Sainath, Eugene Weinstein, Bhuvana Ramabhadran, Yonghui Wu, Ankur Bapna, Zhifeng Chen, and Seungji Lee. 2019. [Large-scale multilingual speech recognition with a streaming end-to-end model](#). In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 2130–2134. ISCA.
- Sarvnaz Karimi, Xiang Dai, Hamed Hassanzadeh, and Anthony N. Nguyen. 2017. [Automatic diagnosis coding of radiology reports: A comparison of deep learning and conventional classification methods](#). In *BioNLP 2017, Vancouver, Canada, August 4, 2017*, pages 328–332. Association for Computational Linguistics.
- Md. Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Huang. 2020a. [Query focused abstractive summarization via incorporating query relevance and transfer learning with transformer models](#). In *Advances in Artificial Intelligence - 33rd Canadian Conference on Artificial Intelligence, Canadian AI 2020, Ottawa, ON, Canada, May 13-15, 2020, Proceedings*, volume 12109 of *Lecture Notes in Computer Science*, pages 342–348. Springer.
- Md. Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Xiangji Huang. 2020b. [WSL-DS: weakly supervised learning with distant supervision for query focused multi-document abstractive summarization](#). In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5647–5654. International Committee on Computational Linguistics.
- Pierre Lison and Jörg Tiedemann. 2016. [Opensubtitles2016: Extracting large parallel corpora from movie and TV subtitles](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).
- Attila Nagy, Bence Bial, and Judit Ács. 2021. Automatic punctuation restoration with bert models. *arXiv preprint arXiv:2101.07343*.
- Askars Salimbajevs. 2016. [Bidirectional LSTM for automatic punctuation restoration](#). In *Human Language Technologies - The Baltic Perspective - Proceedings of the Seventh International Conference Baltic HLT 2016, Riga, Latvia, October 6-7, 2016*, volume 289 of *Frontiers in Artificial Intelligence and Applications*, pages 59–65. IOS Press.
- Ottokar Tilk and Tanel Alumäe. 2015. [LSTM for punctuation restoration in speech transcripts](#). In *INTER-SPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, pages 683–687. ISCA.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.
- Kaituo Xu, Lei Xie, and Kaisheng Yao. 2016. [Investigating LSTM for punctuation prediction](#). In *10th International Symposium on Chinese Spoken Language Processing, ISCSLP 2016, Tianjin, China, October 17-20, 2016*, pages 1–5. IEEE.

Jiangyan Yi, Jianhua Tao, Ye Bai, Zhengkun Tian, and Cunhang Fan. 2020. Adversarial transfer learning for punctuation restoration. *arXiv preprint arXiv:2004.00248*.

Piotr Zelasko, Piotr Szymanski, Jan Mizgajski, Adrian Szymczak, Yishay Carmiel, and Najim Dehak. 2018. [Punctuation prediction model for conversational speech](#). In *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, pages 2633–2637. ISCA.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. [Aligning books and movies: Towards story-like visual explanations by watching movies and reading books](#). In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society.