PunCantonese: A Benchmark Corpus for Low-Resource Cantonese Punctuation Restoration from Speech Transcripts

Yunxiang Li[1], *Pengfei Liu*[2,*], *Xixin Wu*[1], *Helen Meng*[1,2]

[1]The Chinese University of Hong Kong, Hong Kong SAR, China
[2]Centre for Perceptual and Interactive Intelligence, Hong Kong SAR, China

yli@se.cuhk.edu.hk, pfliu@cpii.hk, wuxx@se.cuhk.edu.hk, hmmeng@se.cuhk.edu.hk

Abstract

Punctuation restoration from unsegmented speech transcripts is an essential task to improve the readability of transcripts and can facilitate various downstream NLP tasks. However, there is still lack of systematic studies on punctuation restoration for Cantonese as a low-resource language. This paper introduces a new Cantonese punctuation corpus named PunCantonese, which consists of annotated spoken transcripts and written-style Wikipedia sentences, covering the major punctuations such as ",.?!" and code-switched sentences in Cantonese and English. We also propose a Transformer-based punctuation model which exploits pre-trained multilingual language models, adopts multitask learning for style and punctuation prediction, and introduces a novel Jyutping embedding layer to inject the phonetic features not explicitly available in Cantonese characters. Experimental results show that these methods are effective in improving punctuation restoration, and the Jyutping embedding layer brings an absolute F_1 increase by more than 2%. [1]

Index Terms: punctuation restoration, speech recognition, low-resource language, multitask learning, corpus collection

1. Introduction

Punctuation restoration from unsegmented speech transcripts is essential to improve the readability of speech recognition transcripts and can facilitate various downstream NLP tasks such as named entity recognition, dependency parsing or part-of-speech (POS) tagging. Many corpora and models have been proposed for high-resource languages such as English, Spanish and Mandarin [1–10]. Tilk et al. [11] introduced a bidirectional RNN model with an attention mechanism, which aims to find the relevant parts of the context for punctuation decisions, and evaluated the model on the English and Estonian datasets. Alam et al. [12] explored Transformer-based language models and proposed a data augmentation strategy to simulate typical ASR transcription errors such as *insertion*, *substitution* and *deletion*. Punctuation restoration is also formulated as a sequence labeling task, and tackled with a CRF layer [3, 13] to capture the relationship among punctuations in a token sequence.

As a language-dependent problem, punctuation restoration usually shows poor performance on low-resource languages due to lack of high-quality labeled datasets. There are some research efforts to improve punctuation restoration for the low-resource languages, such as Bangladesh [12], Polish [14], Malay [15], Vietnamese [16], Italian [17], Portuguese [18] and so on. Recently, multilingual approaches [19–23] have been proposed to improve performance for low-resource languages. For example, Li and Lin [19] proposed a 43 language multilingual punctuation model based on LSTM and Byte Pair Encoding (BPE) and showed performance improvement on low-resource languages through fine-tuning the multilingual model. Similarly, Ballesteros and Wanner [21] introduced a transition-based LSTM model using the character-based representations for multilingual punctuation generation.

In this paper, we aim to tackle the low-resource Cantonese punctuation restoration problem from corpus collection to model building. Firstly, we collect a small corpus using the publicly available spoken Cantonese datasets (e.g., Common-Voice), and then crawl the Cantonese Wikipedia, which is in written-form Cantonese. Although a bunch of textual resources like Wikipedia are available, they are mainly written text different from the spoken text in terms of word usages, styles and even grammars. To this end, rather than simply training a corpus with both written text and spoken text indiscriminately, we present a multitask learning framework to predict the style (written and spoken, or formal and informal) of an input text and then predict the punctuation label for each token in the text. Several multitask learning models [4, 24–27] have been proposed for punctuation restoration, e.g., exploiting an extra POS tagging task [24], joint learning with simultaneous speech recognition and punctuation prediction [27]. They typically require additional labels (e.g., POS tags) or parallel speech data with punctuated transcripts, whereas the auxiliary task in our framework is to distinguish the style first and then predict the subsequent punctuations conditioned on the style. Besides, there are code-switched sentences containing both Cantonese and English. Hence, we develop a Transformer-based model based on a pre-trained multilingual language model to deal with both languages and obtain a good initialization of the sentence representation. In addition, we propose to integrate a Jyutping embedding layer into the model to inject the phonetic features from the Jyutping representation of Cantonese characters. The motivation is similar with the paper [28] which proposed to exploit both lexical and acoustic features from word and speech embeddings by a self-attention based model, and thus required both speech and its transcript. However, our approach relies only on the unsegmented transcripts. We summarize the major contributions of this paper as follows:

1. We present a novel corpus named PunCantonese, which poses the real-world challenges for the problem of low-resource Cantonese punctuation restoration.
2. We propose a state-of-the-art Transformer-based model to evaluate PunCantonese and show the effectiveness of Jyutping embeddings, multi-task learning in the proposed model.

*Corresponding author. This research was supported by the Center for Perceptual and Interactive Intelligence (CPII) Ltd under the Innovation and Technology Commission's InnoHK Scheme.

[1]We have open-sourced the corpus and the related source code at https://github.com/cpii-cai/PunCantonese.

2. The PunCantonese Corpus

We introduce a novel corpus named PunCatonese for the task of low-resource Cantonese punctuation restoration. The corpus is compiled from the three data sources, namely Common-Voice[2], PyCantonese[3] and Cantonese Wikipedia[4]. Since it is difficult to obtain a large dataset of spoken Cantonese transcripts with punctuations, we firstly collect a small number of speech transcripts with punctuations from Common-Voice and PyCantonese and then exploit the Cantonese Wikipedia to increase the number of training sentences with punctuations. Consequently, the number of spoken transcripts in our dataset is significantly smaller than the number of written examples from Wikipedia, posing a major challenge for accurately restoring punctuation from different data sources. Nevertheless, our experimental results demonstrate that incorporating a multi-task learning framework allows us to effectively leverage the Wikipedia data and achieve satisfactory performance on the spoken transcripts from Common-Voice and PyCantonese.

Table 1 shows some examples from the PunCantonese corpus, which covers four types of punctuations, namely `COMMA`, `PERIOD`, `QUESTION` and `EXCLAMATION` mark. However, the number of each punctuation type is imbalanced and poses a new challenge to obtain a balanced prediction performance. Besides, it has code-switched examples, which have both English and Cantonese in one single sentence. This calls for a multilingual model to support both Cantonese and English processing.

2.1. Cantonese Speech Transcripts

We obtained the Cantonese speech transcripts with punctuations from the two datasets, namely Common-Voice and PyCantonese. They originally have 8.4k and 37.5K sentences respectively and each sentence ends with an exclamation, question mark, period or no punctuation. During pre-processing, we filtered out the sentences without a punctuation (maybe incomplete sentences) at the end and kept the sentences longer than two words. There are remaining 29.4K sentences in total with 199.1K words. However, these sentences may not be enough to train a reliable punctuation restoration model for Cantonese speech recognition. Therefore, we propose to increase the dataset with the written articles from Cantonese Wikipedia.

2.2. Written Text from Cantonese Wikipedia

We used the publicly available Cantonese Wikipedia to serve as the data for the formal (written) scenarios. Firstly the raw data was divided into sentences, with each sentence on a separate line. To make the dataset clean, we only keep the sentences with a full punctuation, (which means the sentence have a period, or an exclamation or question mark at the end). And the four kinds of punctuations are reserved in our corpus: `COMMA`, `PERIOD`, `QUESTION` and `EXCLAMATION` mark. Note that ":" is replaced to comma, "()[]" are removed with the inside text and other punctuations are simply removed. Any sentence which contains languages other than Cantonese and English are also deleted. Since some sentences are very short (e.g., containing only a single number) or extremely long, we removed all of the sentences with fewer than 8 words or longer than 200 words during the pre-processing step. The pre-processed Wikipedia dataset contains 415.2K sentences with 13.7M words in total.

[2]https://commonvoice.mozilla.org/yue
[3]https://pycantonese.org
[4]https://zh-yue.wikipedia.org

2.3. Corpus Statistics

In Table 2, we present the distributions of the labels in the PunCantonese corpus. In total there are 427.4K sentences, where 50.9K sentences are code-switched containing both Cantonese and English, and the remaining 376.5K sentences contain only Cantonese. The sentences from both Common-Voice and PyCantonese are randomly split into training, validation, and test sets with a ratio of 7:1:2. All sentences from Cantonese Wikipedia are added to the final training set. The average sentence length measured in terms of the number of tokens is 32.

3. Approach

We propose a Transformer-based neural network model to evaluate the PunCantonese corpus. The model exploits pre-trained language models to obtain a good network initialization, and a multi-task learning objective to prevent the network from paying too much attention to the largest subset of written-style Wikipedia sentences rather than the target speech transcripts. Furthermore, we introduce a novel Jyutping embedding layer to represent a Cantonese character with its Jyutping sequence. This potentially enables the model to incorporate phonetic features that are not explicitly available in Cantonese characters.

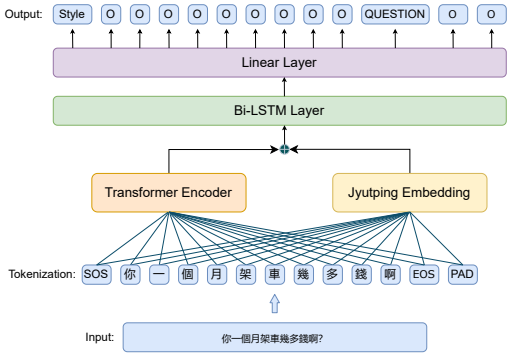

Figure 1: *The Transformer-based model with Jyutping embedding and multitask learning for punctuation restoration.*

3.1. Model Architecture

As illustrated in Figure 1, the proposed model consists of three major components: (1) the pre-trained multilingual language model based on Transformer, (2) the Jyutping embedding layer to inject phonetic features, and (3) a bidirectional LSTM layer followed by a linear layer (activated using `softmax`) for label prediction. First, the input sequence is passed through the Transformer-based multilingual language model and the Jyutping embedding layer, which generates both encoder vectors and Jyutping embeddings. They are added together token by token, and the resulting vectors are fed to the subsequent Bi-LSTM layer to utilize the left and right context for each token. At each time step, the outputs from the forward and backward LSTM layers are concatenated and passed through a single linear layer. The linear layer has 7 output neurons in total, where the first five neurons are used for predicting the four punctuations and the `O` (other) label, and the remaining two neurons for predicting whether the input sequence is formal or informal.

3.2. Pre-trained Language Models

Pre-trained language models [29–32] have shown the state-of-the-art performance on various NLP tasks and are be-

Table 1: *Samples of the PunCantonese Corpus.*

Text	Token	Label	Source	Code-Switch
乜叫做tearing啊？	乜叫做tearing啊	O O O O QUESTION	PyCantonese	TRUE
香港用通貨係港紙。	香港用通貨係港紙	O O O O O O O O PERIOD	Wikipedia	FALSE
走鬼呀借開!	走鬼呀借開	O O O O EXCLAMATION	Common-Voice	FALSE

Table 2: *Basic statistics of the PunCantonese corpus.*

Dataset	Overall	Other(O)	PERIOD	COMMA	QUESTION	EXCLAMATION
Train (Common-Voice + PyCantonese)	99454	84196 (84.7%)	6560 (6.60%)	6515 (6.56%)	1874 (1.88%)	309 (0.31%)
Train (Cantonese Wikipedia)	13797603	12537868 (90.87%)	421779 (3.06%)	835773 (6.06%)	1874 (0.01%)	309 (0.002%)
Validation	14269	12098 (84.8%)	951 (6.66%)	917 (6.43%)	254 (1.78%)	49 (0.34%)
Test	28486	24143 (84.8%)	1895 (6.66%)	1834 (6.44%)	524 (1.84%)	90 (0.32%)
Total	**13840435**	**12574175 (90.8%)**	**424627 (3.07%)**	**838533 (6.06%)**	**2652 (0.02%)**	**448 (0.003%)**

coming the essential cornerstone in deep learning models. It is now a common practice to fine-tune a pre-trained language model for a downstream NLP task. For Cantonese punctuation restoration, we adopt the pre-trained `bert-base-multilingual-uncased`[5] model to support both Cantonese and English in the code-switched sentences[6]. This model was pre-trained using Wikipedia data in 102 languages based on the two objectives of masked language modeling and next sentence prediction.

3.3. Multitask Learning

The PunCantonese corpus is imbalanced in terms of formal and informal sentences due to the large amounts of Wikipedia sentences. Thus the model may tend to overfit on the Wikipedia data, instead of learning from both the Wikipedia data and speech data. To mitigate such problem, we extend the final linear layer in Figure 1 with two additional neurons to predict the style (formal or informal) of a sentence based on the encoder vector of the `SOS` token, together with the other five neurons for punctuation prediction for the subsequent tokens. In our experiments, we observe that the auxiliary style prediction task is effective to improve the performance of punctuation restoration.

3.4. Jyutping Embeddings

Jyutping is the Romanized sequence of a Cantonese character to represent the pronunciation. The steps to generate the Jyutping embeddings for a sentence are as follows: First, we utilize the PyCantonese library to generate the Jyutping sequence for the Cantonese characters in a sentence; Then, we apply an embedding layer from PyTorch on the Jyutping sequence which is padded to a fixed length. Note that we set the output dimension of the embedding layer the same as that of the Transformer encoder output. The obtained Jyutping embeddings are added directly to the output vectors from the Transformer encoder token by token, allowing us to potentially capture both the semantic and the phonetic properties of the Cantonese characters.

4. Experiments

4.1. Experimental Setup

During our experiments, we set the batch size to 32 and the learning rate for all the models as 2e-5. Each model was trained for a total of 15 epochs and the best model (epoch) was chosen based on the validation set. The dimension of the LSTM was set the same with that of the language model output. Additionally, we set the maximal sequence length as 128 and kept the random seed fixed as 0. To alleviate the class imbalance problem, the model was trained based on the focal loss proposed by [33]. Following the settings in [33], we chose $\gamma = 2$ and $\alpha = 0.25$ and 0.75 for easy and hard examples, respectively.

4.2. Evaluation Metrics

Following [1, 24], we chose precision (P), recall (R) and F_1 as the evaluation metrics to calculate per-class and overall performance, ignoring the `O` predictions in the calculation, as shown below for the `COMMA` punctuation:

$$P = \frac{\text{\#correctly predicted } \texttt{COMMA}}{\text{\#all predicted } \texttt{COMMA}} \tag{1}$$

$$R = \frac{\text{\#correctly predicted } \texttt{COMMA}}{\text{\#all actual } \texttt{COMMA}} \tag{2}$$

$$F_1 = \frac{2 * P * R}{P + R} \tag{3}$$

We also report macro-$F1$, which is the average F_1 score of the four punctuations, since our dataset is imbalanced and we want to treat each punctuation equally.

4.3. Experimental Results

As shown in Table 3, the multilingual pre-trained language model (`bert-multilingual-base-uncased`) is a strong baseline method even using only the small spoken transcripts for training. Adding the Wikipedia sentences mainly improves the PERIOD punctuation, while multitask learning improves the baseline on all the punctuations, particularly on EXCLAMATION. Similarly, Jyutping embeddings outperform the baseline model on all the punctuations.

We also observe the performance variations across different punctuations. Specifically, the F_1-score for `PERIOD` was consistently above 90% for all models, followed by the question mark, while both the comma and exclamation mark exhibited F_1-scores lower than 70%. One possible explanation for this disparity is that classifying `PERIOD` is relatively straightforward, as the semantic differences between sentences make them easier to separate. In contrast, classifying `COMMA` is a more ambiguous and challenging task because their usage can be optional in certain contexts. Although the QUESTION mark has a relatively smaller number of training examples, it often follows with specific words or phrases (e.g., 嗎, 咩), leading to relatively higher performance.

[5]https://huggingface.co/bert-base-multilingual-uncased

[6]We also tried the XLM-Roberta model (pre-trained on CommonCrawl data containing 100 languages) which however gives worse performance than bert-base-multilingual-uncased on PunCantonese.

Table 3: *Experimental results on the PunCantonese corpus.*

Method	PERIOD			COMMA			QUESTION			EXCLAMATION			OVERALL			
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	**macro-F_1**
bert-base-uncased	77.1	**99.3**	86.8	41.7	2.4	4.5	72.7	3.0	5.9	82.2	41.1	54.8	75.7	45.6	56.9	38.0
bert-base-multilingual-uncased	**91.8**	93.9	92.8	69.8	50.4	58.5	77.1	**75.4**	76.3	82.8	53.3	64.9	82.1	72.5	76.9	73.1
+ Wiki	90.3	95.7	93.0	66.3	60.2	63.1	82.1	71.0	76.2	89.8	48.9	63.3	79.8	76.8	78.3	73.9
+ Wiki + Multitask	90.1	97.5	**93.6**	71.1	57.9	63.8	**87.5**	68.1	76.6	95.9	52.2	**67.6**	82.8	76.3	79.4	75.4
+ Wiki + Jyutping	90.2	96.6	93.3	**75.3**	**65.4**	**70.0**	84.4	70.2	**76.7**	**97.6**	45.6	62.1	**83.9**	**79.2**	**81.5**	75.5
+ Wiki + Jyutping + Multitask	91.5	94.7	93.1	73.7	63.4	68.2	78.9	74.2	76.5	92.6	**55.6**	66.4	83.1	78.2	80.6	**77.2**

5. Discussions

Effect of Pre-trained Language Models. Firstly, we investigate whether there exists a performance gap between pre-trained monolingual models and multilingual models using the spoken transcripts without Wikipedia data. As shown in Table 3, we conducted experiments using `bert-base` and `bert-multilingual-base-uncased`. Our findings indicate that the latter model with all settings outperforms the former one by approximately 20% in terms of the overall F_1 score. This could be attributed that the monolingual model, which are pre-trained entirely on English, has no Cantonese characters in its vocabulary, resulting in a significant negative impact on contextual understanding. As a result, multilingual models are more suitable to deal with code-switched sentences with both Cantonese and English, which are commonly observed in Cantonese speech. Hence, we adopted only the `bert-multilingual-base-uncased` model for the subsequent experiments.

Effect of Adding Cantonese Wikipedia. After comparing the results of the models trained on the dataset with and without Wikipedia data, we have observed an overall increase of approximately 2% in the F_1 score, with a more significant increase of 5% in the F_1 score for PERIOD. These results suggest that incorporating written text can facilitate the model in accurately recognizing punctuations. It is supposed that a larger corpus of written text may help the model in generating more effective embeddings to capture the semantics and identify sentence boundaries.

Effect of Multitask Learning. Comparing the baseline (`bert-multilingual-base-uncased`) with the model using multitask learning in Table 3, we observe an increase in the F_1 scores for all the four punctuations, along with a 2% increase in the overall F_1 score. We hypothesize that the baseline model might be prone to over-fitting to the written-style Wikipedia sentences, which constitutes the largest subset of the PunCantonese corpus. As shown in Table 3, the major performance gain was observed in `EXCLAMATION`, which supports our hypothesis because `EXCLAMATION` is frequently used in spoken language but is relatively uncommon in written sources. By introducing the multitask objective, the model needs to pay more attention to learning to recognize the style of each sentence at the beginning, preventing it from becoming overly reliant on the largest proportion of Wikipedia sentences.

Effect of Jyutping Embedding. The application of Jyutping embedding in our experiment yielded a notable improvement in model performance, with an increase of approximately 4% in total F_1 and an increase in F_1 for all the four punctuation marks as compared to the baseline. By integrating Jyutping embedding into our model architecture, we were able to capture both the semantic and phonetic information of the text more effectively, resulting in improved performance across all four punctuation marks. Moreover, Cantonese characters often have multiple pronunciations, and different characters can share the same pronunciation. Jyutping, however, offers a standardized and consistent means of representing the pronunciation of Cantonese words. Incorporating Jyutping embeddings into the model could also potentially enhance its ability to generalize to out-of-vocabulary or unseen words. These findings suggest that Jyutping embedding can serve as an effective technique for boosting the performance of natural language processing models in Cantonese and other tonal languages.

Limitations and Error Analysis. We plot the confusion matrix of the model with Jyutping embeddings and multitask learning on the test set to analyze the errors and model limitations, as shown in Figure 2. It can be seen that the model still mis-classifies both "Exclamation" and "Question" as "Period", and "Comma" as "Other". This indicates the remaining challenges in distinguishing sentence segments and tones using only the text for punctuation restoration, and we leave them as future work for further investigation.

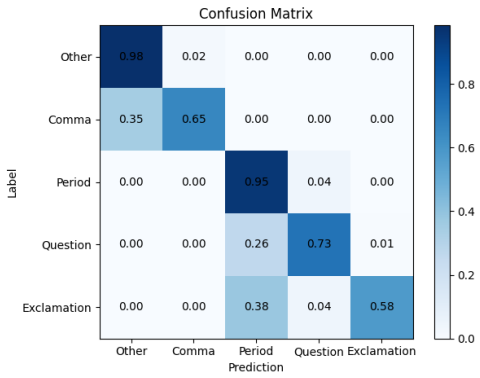

Figure 2: *The confusion matrix of the model with Jyutping embeddings and multitask learning on the test set.*

6. Conclusion

This paper presents a novel corpus named PunCantonese for the problem of low-resource Cantonese punctuation restoration, which is essential to improve the readability of ASR transcripts. PunCantonese is collected from annotated speech transcripts and written-style Cantonese Wikipedia to cover punctuations in both spoken and written scenarios, as well as code-switched sentences commonly used in Cantonese.

We introduce a Transformer-based neural network model which adopts a multilingual pre-trained language model for code-switched sentences, and a novel Jyutping embedding layer to integrate the phonetic information of Cantonese characters, as well as a multitask learning objective to discriminate speech transcripts and written text. Experimental results on the PunCantonese corpus show that the model achieves the state-of-the-art performance, particularly brought by the Jyutping embedding layer thanks to its additional phonetic information.

7. 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] O. Klejch, P. Bell, and S. Renals, "Punctuated transcription of multi-genre broadcasts using acoustic and lexical approaches," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 433–440.

[3] 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.

[4] A. Vāravs and A. Salimbajevs, "Restoring punctuation and capitalization using transformer models," in *Statistical Language and Speech Processing: 6th International Conference, SLSP 2018, Mons, Belgium, October 15–16, 2018, Proceedings 6*. Springer, 2018, pp. 91–102.

[5] 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.

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

[7] Q. Huang, T. Ko, H. L. Tang, X. Liu, and B. Wu, "Token-level supervised contrastive learning for punctuation restoration," in *Annual Conference of the International Speech Communication Association*. International Speech Communication Association, 2021.

[8] Q. Chen, W. Wang, M. Chen, and Q. Zhang, "Discriminative self-training for punctuation prediction," in *INTERSPEECH*, 2021.

[9] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos *et al.*, "The ami meeting corpus," in *Proceedings of Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research*. Noldus Information Technology, 2005, pp. 137–140.

[10] M. Cettolo, C. Girardi, and M. Federico, "Wit3: Web inventory of transcribed and translated talks," in *Proceedings of the Conference of European Association for Machine Translation (EAMT)*, 2012, pp. 261–268.

[11] O. Tilk and T. Alumäe, "Bidirectional recurrent neural network with attention mechanism for punctuation restoration." in *INTERSPEECH*, 2016, pp. 3047–3051.

[12] 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)*, 2020, pp. 132–142.

[13] W. Gale and S. Parthasarathy, "Experiments in character-level neural network models for punctuation." in *INTERSPEECH*, 2017, pp. 2794–2798.

[14] M. Pogoda and T. Walkowiak, "Comprehensive punctuation restoration for english and polish," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 4610–4619.

[15] A. K. Rao, H. Thi-Nga, and C. E. Siong, "Punctuation restoration for singaporean spoken languages: English, malay, and mandarin," *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 546–552, 2022.

[16] H. T. T. Uyen, N. A. Tu, and T. D. Huy, "Vietnamese capitalization and punctuation recovery models," in *INTERSPEECH*, 2022.

[17] A. Miaschi, A. A. Ravelli, and F. Dell'Orletta, "Punctuation restoration in spoken italian transcripts with transformers," in *AIxIA 2021–Advances in Artificial Intelligence: 20th International Conference of the Italian Association for Artificial Intelligence, Virtual Event, December 1–3, 2021, Revised Selected Papers*. Springer, 2022, pp. 245–260.

[18] T. B. D. Lima, P. Miranda, R. F. Mello, M. Wenceslau, I. I. Bittencourt, T. D. Cordeiro, and J. José, "Sequence labeling algorithms for punctuation restoration in brazilian portuguese texts," in *Intelligent Systems: 11th Brazilian Conference, BRACIS 2022, Campinas, Brazil, November 28–December 1, 2022, Proceedings, Part II*. Springer, 2022, pp. 616–630.

[19] X. Li and E. Lin, "A 43 language multilingual punctuation prediction neural network model." in *INTERSPEECH*, 2020, pp. 1067–1071.

[20] N. M. Guerreiro, R. Rei, and F. Batista, "Towards better subtitles: A multilingual approach for punctuation restoration of speech transcripts," *Expert Systems with Applications*, vol. 186, p. 115740, 2021.

[21] M. Ballesteros and L. Wanner, "A neural network architecture for multilingual punctuation generation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing; 2016 Nov. 1-5; Austin (TX, USA).[place unknown]: ACL; 2016. p. 1048-53*. ACL (Association for Computational Linguistics), 2016.

[22] V. Chordia, "Punktuator: A multilingual punctuation restoration system for spoken and written text," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, 2021, pp. 312–320.

[23] M. Hentschel, E. Tsunoo, and T. Okuda, "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)*. IEEE, 2021, pp. 7773–7777.

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

[25] M. Dixit and S. B. K. Kirchhoff, "Robust prediction of punctuation and truecasing for medical asr," *ACL 2020*, p. 53, 2020.

[26] R. Pappagari, P. Żelasko, A. Mikołajczyk, P. Pezik, and N. Dehak, "Joint prediction of truecasing and punctuation for conversational speech in low-resource scenarios," *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1185–1191, 2021.

[27] Z. Zhou, T. Tan, and Y. Qian, "Punctuation prediction for streaming on-device speech recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7277–7281.

[28] 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.

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

[30] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Annual Meeting of the Association for Computational Linguistics*, 2019.

[31] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[32] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[33] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 318–327, 2017.