An Empirical Study on Punctuation Restoration for English, Mandarin, and Code-switching Speech

Liu Changsong, Ho Thi Nga, and Chng Eng Siong

Nanyang Technological University, Singapore
{liuc0062,ngaht,aseschng}@ntu.edu.sg

Abstract. Punctuation restoration is a crucial task in enriching automated transcripts produced by Automatic Speech Recognition (ASR) systems. This paper presents an empirical study on the impact of employing different data acquisition and training strategies on the performance of punctuation restoration models for multilingual and codeswitching speech. The study focuses on two of the most popular Singaporean spoken languages, namely English and Mandarin in both monolingual and codeswitching forms. Specifically, we experimented with in-domain and out-of-domain evaluation for multilingual and codeswitching speech. Subsequently, we enlarge the training data by sampling the codeswitching corpus by reordering the conversational transcripts. We also proposed to ensemble the predicting models by averaging saved model checkpoints instead of using the last checkpoint to improve the model performance. The model employs a slot-filling approach to predict the punctuation at each word boundary. Through utilizing and enlarging the available datasets as well as ensemble different model checkpoints, the result reaches an F1 score of **76.5%** and **79.5%** respectively for monolingual and codeswitch test sets, which exceeds the state-of-art performance. This investigation contributes to the existing literature on punctuation restoration for multilingual and code-switch speech. It offers insights into the importance of averaging model checkpoints in improving the final model's performance. Source codes and trained models are published on our Github's repo[1] for future replications and usage.

Keywords: Punctuation Restoration · Multilingual · Codeswitching · Automatic Speech Recognition · Singaporean Speech.

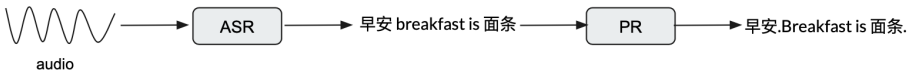
1 Introduction

Punctuation is an essential aspect of language since it helps to convey meaning and structure clearly. As shown in Figure 1, an ASR system first takes in raw audio files and outputs transcripts. However, transcripts generated from most ASR systems generally remain unpunctuated, resulting in significant difficulties for human and machine processing of the transcripts. This includes reducing

[1] https://github.com/charlieliu331/Punctuation_Restoration

readability, impacting comprehension, as well as disturbing and deteriorating downstream tasks' performance [1], such as machine translation [2], sentence dependency parsing [3], and sentiment analysis [4]. Hence, we need Punctuation Restoration (PR) to address this problem. Some existing approaches treat PR as a sequence labeling task [5], while others treat it as a slot-filling task [6], which is what we employ in this paper. Our objective is to investigate the impact of using in-domain and out-domain datasets, and model averaging on the multilingual and codeswitching punctuation restoration model in the speech transcript through an empirical study. We employed the same network architecture proposed by [6]. Our motivation is to produce a punctuation restoration model that can punctuate codeswitching speech, to provide insights on the impact of data preparation and model ensemble on the performance of the model, and thus to advance the state-of-the-art. This in turn will contribute to cross-cultural and cross-language communications and the performance of downstream tasks. We will present our method and analyze the impact of using in-domain and out-of-domain data, strategies for data acquisition, and model ensemble on the performance of a punctuation restoration model in subsequent sections.

Fig. 1. ASR process diagram.



2 Related Work

Restoring punctuations for speech transcripts generally involves utilizing prosodic features and lexical features. Prosodic features, derived from audio, consist of characteristics such as pause duration, and fundamental frequency(F0). There are studies like [7] and [8] that use LSTM to punctuate texts with prosodic features including pause durations. Subsequent studies, such as [9], have modified the approach by replacing LSTM with bidirectional LSTM cells and incorporating an attention mechanism. However, using prosodic features can be problematic due to individual deviations in speech patterns such as tone and pitch. Lexical features, on the other hand, have been seen to be the sole input sources to various models such as HMM [10], Conditional Random Field (CRF) [11], word embedding [12], and attention-based deep learning[13]. These lexical-based approaches are commonly easier to implement than prosody-based methods. This is because they only require processing text inputs and do not require aligning audio with text. Some studies have shown that combining both prosodic and lexical features can yield the best performance. For example, in [14], the authors combined both lexical and prosodic features by integrating word-based

and prosody-based networks to compensate for the deficiencies of each approach. Within the scope of this paper, we employed only lexical features acquired from texts for simplification.

For the task of multilingual punctuation restoration, recent research has focused on the utilization of multilingual language models (MLM), including Multilingual T5 [15], Multilingual-BERT [16], and XLM-RoBERTa [17], etc. These models can generate multilingual embeddings that are customizable for a wide range of subsequent tasks, including punctuation restoration. They come as pretrained and are suitable for natural language processing (NLP) tasks. Incorporating MLMs into punctuation restoration can provide a more robust and accurate approach to handling diverse language varieties, code-switching, and multilingual texts. Previous works such as [16] and [17] are able to handle multilingual punctuation restoration, however, to our best knowledge there is no existing work in handling code-switching speech.

3 Methodology

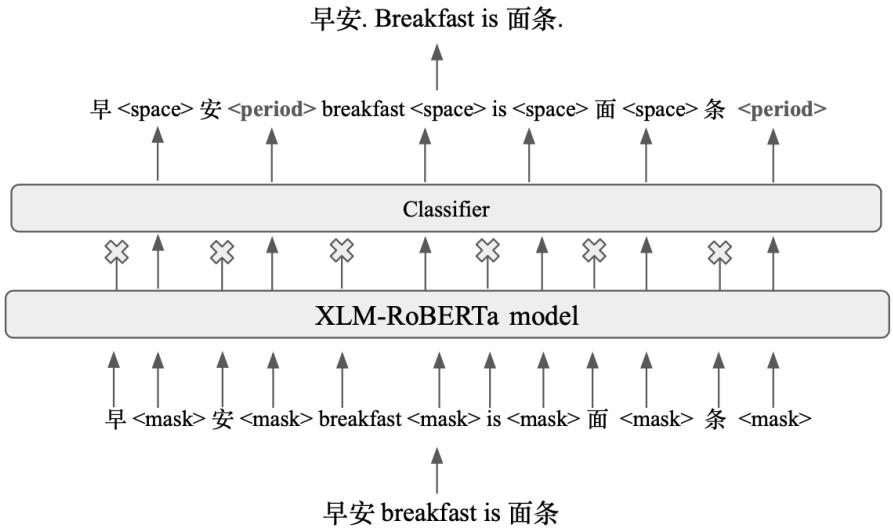
3.1 Network Architecture

This work employs the off-the-shelf model architecture proposed by [6] for multilingual punctuation which consists of a pre-trained XLM-RoBERTa (XLM-R) model followed by a classifier. The architecture is illustrated in Figure 2. The XLM-R model generates multiple embeddings in a self-supervised way and contains multiple self-attention transformer-encoder layers. The input data is tokenized using the multilingual tokenizers, i.e. SentencePiece and Jieba, and subsequently the <mask> token is added after every word. The classifier comprising two full-connected (FC) layers predicts punctuation for each masked token. The output contains tokens that represent commas, periods, or spaces. To explore the impact of data preparation and acquisition on training a Punctuation Restoration (PR) model, only this network architecture has been employed to train different PR models using datasets that have been prepared and acquired in different ways.

3.2 Datasets

Multilingual and Codeswitching Datasets We aim to build a punctuation restoration model that can handle both multilingual speech, i.e. speakers either speak English or Mandarin, and code-switch, i.e. speakers speech English and Mandarin interchangeably either within a conversation or even within a sentence. To serve this purpose we utilized the IWSLT2012 TED Talks dataset and our in-house English-Mandarin codeswitching dataset. The IWSLT2012 TED Talk dataset, named IWSLT2012 or TED, is the same as the dataset used in [6, 13]. IWSLT2012 contains the English transcripts extracted from EN-FR parallel machine translation tracks and Mandarin transcripts extracted from the ZH-EN parallel tracks. IWSLT2012 is considered a multilingual dataset in which its

Fig. 2. Model architecture diagram.



topic of discussion are varied and its speakers are from anywhere in the world. The in-house dataset is recorded by our team with recording devices and manually transcribed by human transcribers, we named it NTU-EnMan or EnMan in short. NTU-EnMan contains conversational speech between two Singaporeans. The speakers may speak in English, Mandarin, or codeswitch between the two languages naturally during the entire conversation. The topics of discussion varied from daily life conversations such as hobbies, personal opinions on specific events, or road direction. The conversation between any pair of speakers was recorded in separate audio channels and transcribed accordingly into separate transcript files.

Data Acquisition With regard to enlarging the training dataset, we acquire additional variations of NTU-EnMan by employing two preprocessing strategies for the NTU-EnMan dataset. Specifically, we first combine all the transcripts from each speaker in conversational order, however, we let the interrupting speech be kept ordered according to its start time. Meaning in a conversation between speaker A and speaker B if speaker A has not finished her sentence but speaker B interrupted, we keep the conversation order as it is. For the second strategy, we kept transcripts from speaker A's and speaker B's speeches separate. That means the conversational context between speaker A and speaker B has been lost. We named this extra set of data sampling NTU-EnMan-extra and used it together with TED and NTU-EnMan for the model training. Overall, our experiments on data preprocessing and combination aim to comprehensively evaluate

the performance of our model under different data conditions, including various combinations of languages, conversation structures, and conversation orderings.

3.3 Models and Ensemble Model

Baseline Models We established our baselines by employing the network architecture described in 3.1 to train punctuation restoration models for multilingual and codeswitching speech. The models are trained separately on TED and NTU-EnMan datasets. Results are reported on evaluation sets from both datasets to see the impact of using in-domain and out-of-domain for training punctuation restoration models. These two models stand as baselines for further experiments.

Multilingual and Codeswitch Model To enable the capacity of predicting both English, Mandarin, and English-Mandarin codeswitching speech, we used both TED and NTU-EnMan for model training.

Model Ensemble Conventionally, the best PR model is usually selected at the training checkpoint that gains the best results on development sets. In this work, we tried to average multiple model checkpoints to yield an overall better model. We first save the checkpoint of the model at each training epoch in every training experiment. We then compare the performance of each training epoch on development sets to choose the top-performance checkpoints. These checkpoints are then averaged to further improve the final models.

4 Experiments

In this section, details of the datasets are described, and the experimental setup and results are presented and discussed.

4.1 Data Preprocessing

Dataset preprocessing is a critical step in preparing data for machine learning models. In this section, we describe the preprocessing steps we used both IWSLT2012 and NTU-EnMan datasets.

Text Normalization For NTU-EnMan, which is stored in TextGrid files, we first extract the text from each file and combine the resulting text files into a single complete text corpus. We then perform text normalization to remove unnecessary symbols such as brackets, "–EMPTY–", and non-speech noises. For the IWSLT2012 dataset, we used dual-lingual datasets prepared by [6], in which separate English and Chinese tracks are combined. For all the mentioned datasets, punctuations are converted or removed to match three classes: period, comma, and space. We convert question marks ('?') and exclamation marks ('!') to periods ('.') to indicate the end of a sentence.

Dataset Splitting After preprocessing the datasets, we split them into training, validation, and test sets. For the IWSLT2012 dataset, we used the provided train, validation, and test splits. For NTU-EnMan, we chose an 80:10:10 split for train, validation, and test sets. It's worth noting that when splitting the data into train, validation, and test sets, we performed the splitting in a conversation-based manner to ensure that the model learns to predict punctuation within the context of a complete conversation. To provide insights into the characteristics of the preprocessed datasets, we present statistics on the number of words, periods, and commas in each dataset in Table 1.

Table 1. Number of words, commas, and periods in train, valid, and test sets.

Dataset	train			valid			test		
	word	comma	period	word	comma	period	word	comma	period
IWSLT2012	4.6M	350K	145K	36.4K	2.5K	1.5K	47.7K	3.3K	2.2K
NTU-EnMan	1.1M	67.4K	82K	144K	9.9K	10.3K	132K	8.5K	10K

Overall, our dataset preparation process aims to standardize the datasets, ensure they are suitable for machine learning models and provide a consistent evaluation methodology across all datasets. These statistics can be used to assess the distribution of punctuation in the datasets and to identify potential biases or imbalances in the data.

4.2 Experiment Setup

The experiments were set with parameters presented as follows:

- The network architecture was modified so that the classifier comprises two linear layers with dimensions of 1568 and 3, respectively. This is to reduce the output layer from 4 to 3 which corresponds to the prediction of 3 output classes, i.e. periods, commas, and spaces.
- Learning rates were set at 1e-4 for the classification layers and 3e-5 for the XLM-R model which follows the setting in the original model [6].
- For loss function we use Negative Log-Likelihood Loss, a commonly used loss function, particularly for classification tasks.
- We use the RAdam [18] optimizer, and a warm-up phase with 300 warm-up steps to gradually increase the learning rate from a small value to the optimal value.
- The maximum number of epochs was set at 10 and batch size was set at 4.
- Precision, Recall, and F1-score metrics were used to evaluate the models' performance on predicting three punctuation classes: period, comma, and space.

4.3 Results and Discussion

Multilingual and codeswitching models To establish baselines for the model's performance, we initiated training it on each dataset. Subsequently, we train the model using the combination of both datasets expecting that the model is able to predict punctuations for both English, Mandarin and English-Mandarin codeswitching speech. The results are presented in Table 2 for models trained at 1 and 5 epochs. We named the models by their corresponding data sets, i.e. TED, EnMan and TED + EnMan. It can be seen that among the baselines models, which is TED and EnMan, the models performed well on their corresponding validation sets, however, degraded significantly when tested with the other validation set. For example, looking at the results when models were trained at 1 epoch, TED model achieved 74.9% F1-score when being tested with TED validation set, however, it only achieved 66.3% on EnMan validation set, resulting in 8.3% absolute reduction. The gap is even larger for the EnMan model, where the reduction was 25.2% from 77.0% to 51.8% in F1-score. This is possible because the size of training data in EnMan is significantly more than TED, and also EnMan dataset contains codeswitching sentences that further confused the model. The results were further biased when the models trained at 5 epochs, in which the gap between F1-score on TED and EnMan validation sets when tested on the TED model increased to 11.4% and 28.2% with EnMan model. These gaps, however, were reduced significantly when both the data were used to train TED+EnMan model. Performance of the model at 1 epoch was 77.3% on EnMan val., and 74.2% on TED val., and at 5 epochs was 78.6% and 74.4% respectively. It is also interesting to see that the best performance on EnMan val. was achieved at 5 epochs of training for TED+EnMan model, however, the model performance on TED val. is slightly reduced, from 74.9% with TED model to 74.4% with TED+EnMan model.

Table 2. Performance of baseline models and codeswitching model.

Models	**1 epoch**		**5 epoch**	
	PR/RC/F1	**PR/RC/F1**	**PR/RC/F1**	**PR/RC/F1**
	TED Val.	**EnMan Val.**	**TED Val.**	**EnMan Val.**
TED	**71.4/78.8/74.9**	59.8/74.5/66.3	**70.8/79.4/74.9**	55.8/73.8/63.5
EnMan	46.4/63.0/51.8	74.2/81.0/77.0	43.9/61.4/50.4	**76.3/81.1/78.6**
TED + EnMan	70.0/79.0/74.2	**75.2/79.5/77.3**	70.9/78.2/74.4	76.0/81.3/78.6

Data Acquisition Next, we add NTU-EnMan-extra to the training set, and retrain the model resulting in a model name TED+EnMan+EnMan-extra. Results of this model in comparison to results of TED+EnMan model are presented in Table 3. Unfortunately, the results showed that the model with extra data from NTU-EnMan-extra does not help to improve the model performance on the validation sets. In fact, there were slight reductions in most of the cases, except on

the EnMan val. when tested on the TED+EnMan+EnMan-extra model trained at 1 epoch. This probably suggested that acquiring additional training data by resampling or re-ordering the existing data doesn't help in improving the model performance.

Table 3. Performance on two datasets combined.

Models	1 epoch		5 epochs	
	PR/RC/F1	PR/RC/F1	PR/RC/F1	PR/RC/F1
	TED Val.	**EnMan Val.**	**TED Val.**	**EnMan Val.**
TED+EnMan	**70.0/79.0/74.2**	75.2/79.5/77.3	**70.9/78.2/74.4**	**76.0/81.3/78.6**
TED+EnMan+ EnMan-extra	70.6/78.0/74.0	**75.4/81.7/78.1**	69.6/78.4/73.7	76.0/80.9/78.4

Model Ensemble

Performance of models at one to ten epochs We extended the training of models for single and combined datasets up to 10 epochs. Figure 3 shows the performance results of each model with results reporting on the TED val. EnMan val. and weighted F1-score combining TED val. + EnMan val. It can be seen TED model performs best at the 5th epoch, EnMan performs best at the 4th epoch and TED+EnMan performs best at the 3rd epoch. However, results are not much different from the 3rd epoch to the 7th epoch on all models. It seems that after 7 epochs, the performance of all models shows a tendency of degrading because of overfitting.

Performance of averaged models Inspired by work reported in [18], the best model is formed by averaging a selected range of training epochs, we found that this applied to the punctuation restoration task, too. Our empirical experiments showed that averaging from the 4th epoch to the 6th epoch produced the best results on TED, EnMan, and TED+EnMan models. Table 4 shows the F1-score obtained on the best-performing model from the 1st to 10th epochs, models that were averaged from the 4th to 6th epochs, and models that were averaged from the 5th to 7th epochs. Similar results were observed when we evaluated these models on test sets from TED and EnMan datasets as showed in Table 5. This confirms our hypothesis that ensemble model works best for the task. Our final model averaged from epochs 4th to 6th achieved **75.8%** and **76.0%** on the TED validation and test sets, which is higher than results reported on [6]. Furthermore, our model has the capacity of predicting punctuations for codeswitching speech with F1 scores of **78.8%** for validation set and **78.4%** for test set, which significantly outperforms the model that only trained on English, Mandarin multilingual speech corpus, i.e. TED.

Fig. 3. F1-score on evaluation sets when trained models up to 10 epochs

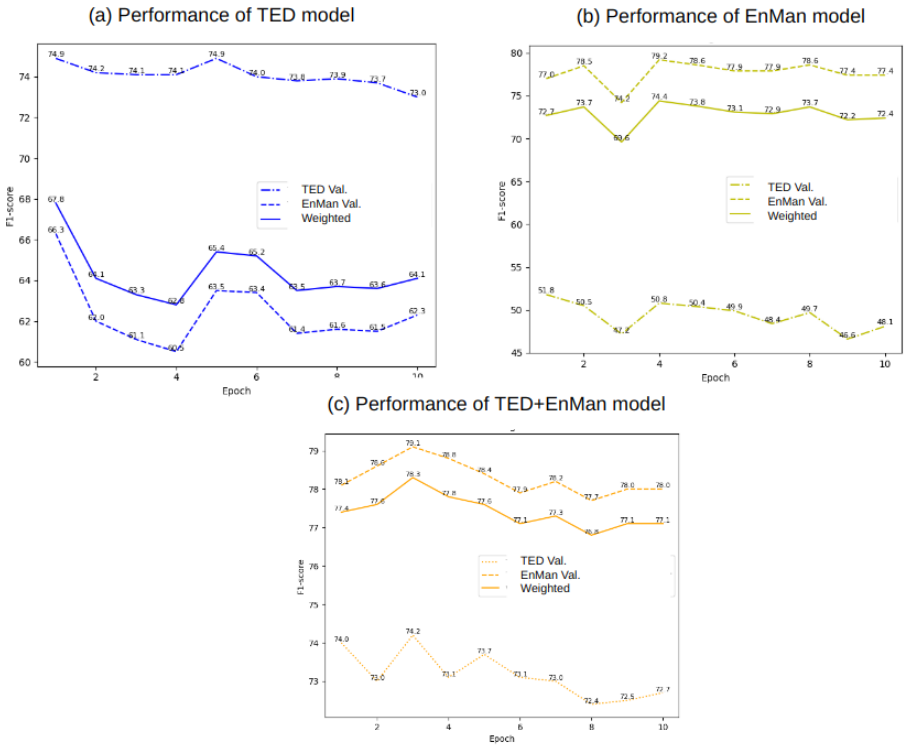


Table 4. Performance best epoch and averaged epochs on validation sets.

Models	**best epoch**		**averaged 4-5-6**		**averaged 5-6-7**	
	TED Val.	**EnMan Val.**	**TED Val.**	**EnMan Val.**	TED Val.	**EnMan Val.**
TED	**74.9**	66.3	75.3	63.2	74.9	63.4
EnMan	50.8	**79.2**	51.6	**79.5**	51.1	**79.5**
TED + EnMan	74.4	78.6	**75.8**	78.8	**75.4**	79.1

Table 5. Performance best epoch and averaged epochs on test sets.

Models	**best epoch**		**averaged 4-5-6**		**averaged 5-6-7**	
	TED Test	**EnMan Test**	**TED Test**	**EnMan Test**	**TED Test**	**EnMan Test**
TED	74.3	66.1	**76.5**	63.2	**76.1**	63.5
EnMan	52.5	**78.3**	53.2	**78.7**	52.9	78.4
TED + EnMan	**75.2**	78.1	76.0	78.4	75.6	**78.5**

5 Conclusion and Future work

In this study, we have investigated the importance of using the right dataset and checkpoint combinations for improving the performance of a multilingual and codeswitching punctuation restoration model. Our study offers insights into the challenges and opportunities of working with multilingual and codeswitching transcripts, highlighting the importance of developing tools and technologies that can support communication across languages and cultures. We believe that the methodology and findings presented in this paper can be applied to other areas of NLP. While our investigation has shown promising results, there are several limitations and areas for future research to consider. For example, we can explore different model architectures or incorporate additional datasets. This could yield further improvements in performance. Additionally, further research can be conducted on more languages. Overall, our study contributes to the existing literature on punctuation restoration and demonstrates the importance of dataset combination and epoch selection for improving model performance. We hope that our findings will inspire further research in this area, supporting the development of more accurate and efficient punctuation restorations.

Acknowledgements This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-100E-2022-102). We would like to acknowledge the High Performance Computing Centre of Nanyang Technological University Singapore, for providing the computing resources, facilities, and services that have contributed significantly to this work.

References

1. Piotr Żelasko, Piotr Szymański, Jan Mizgajski, Adrian Szymczak, Yishay Carmiel, and Najim Dehak. Punctuation prediction model for conversational speech. *arXiv preprint arXiv:1807.00543*, 2018.
2. Stephan Peitz, Markus Freitag, Arne Mauser, and Hermann Ney. Modeling punctuation prediction as machine translation. In *Proceedings of the 8th International Workshop on Spoken Language Translation: Papers*, pages 238–245, 2011.
3. Valentin I Spitkovsky, Hiyan Alshawi, and Dan Jurafsky. Punctuation: Making a point in unsupervised dependency parsing. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 19–28, 2011.
4. Attila Nagy, Bence Bial, and Judit Ács. Automatic punctuation restoration with bert models. *arXiv preprint arXiv:2101.07343*, 2021.
5. Tiago B De Lima, Pericles Miranda, Rafael Ferreira Mello, Moesio Wenceslau, Ig Ibert Bittencourt, Thiago Damasceno Cordeiro, and Jário 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*, pages 616–630. Springer, 2022.

6. Abhinav Rao, Ho Thi-Nga, and Chng Eng Siong. Punctuation restoration for singaporean spoken languages: English, malay, and mandarin. In *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 546–552. IEEE, 2022.
7. Fernando Batista, Diamantino Caseiro, Nuno Mamede, and Isabel Trancoso. Recovering punctuation marks for automatic speech recognition. In *Eighth Annual Conference of the International Speech Communication Association*, 2007.
8. Ottokar Tilk and Tanel Alumäe. Lstm for punctuation restoration in speech transcripts. In *Sixteenth annual conference of the international speech communication association*, 2015.
9. Ottokar Tilk and Tanel Alumäe. Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In *Interspeech*, pages 3047–3051, 2016.
10. Madina Hasan, Rama Doddipatla, and Thomas Hain. Multi-pass sentence-end detection of lecture speech. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.
11. Wei Lu and Hwee Tou Ng. Better punctuation prediction with dynamic conditional random fields. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 177–186, 2010.
12. Xiaoyin Che, Cheng Wang, Haojin Yang, and Christoph Meinel. Punctuation prediction for unsegmented transcript based on word vector. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 654–658, 2016.
13. Karan Makhija, Thi-Nga Ho, and Eng-Siong Chng. Transfer learning for punctuation prediction. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 268–273. IEEE, 2019.
14. György Szaszák and Máté Akos Tündik. Leveraging a character, word and prosody triplet for an asr error robust and agglutination friendly punctuation approach. In *INTERSPEECH*, pages 2988–2992, 2019.
15. Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020.
16. Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*, 2019.
17. Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.
18. Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.