

# Multimodal Semi-supervised Learning Framework for Punctuation Prediction in Conversational Speech

Monica Sunkara, Srikanth Ronanki, Dhanush Bekal, Sravan Bodapati, Katrin Kirchhoff

Amazon AWS AI

{sunkara1, ronanks}@amazon.com

## Abstract

In this work, we explore a multimodal semi-supervised learning approach for punctuation prediction by learning representations from large amounts of unlabelled audio and text data. Conventional approaches in speech processing typically use forced alignment to encoder per frame acoustic features to word level features and perform multimodal fusion of the resulting acoustic and lexical representations. As an alternative, we explore attention based multimodal fusion and compare its performance with forced alignment based fusion. Experiments conducted on the Fisher corpus show that our proposed approach achieves  $\sim 6\text{-}9\%$  and  $\sim 3\text{-}4\%$  absolute improvement (F1 score) over the baseline BLSTM model on reference transcripts and ASR outputs respectively. We further improve the model robustness to ASR errors by performing data augmentation with N-best lists which achieves up to an additional  $\sim 2\text{-}6\%$  improvement on ASR outputs. We also demonstrate the effectiveness of semi-supervised learning approach by performing ablation study on various sizes of the corpus. When trained on 1 hour of speech and text data, the proposed model achieved  $\sim 9\text{-}18\%$  absolute improvement over baseline model.

**Index Terms:** speech recognition, punctuation prediction, multimodal fusion, semi-supervised learning

## 1. Introduction

The output text generated from automatic speech recognition (ASR) systems is typically devoid of punctuation and sentence formatting. Lack of sentence segmentation and punctuation makes it difficult to comprehend the ASR output. For example, consider the two sentences: "Let's eat Grandma" vs. "Let's eat, Grandma!". Punctuation restoration not only helps understand the context of the text but also greatly improves the readability. Punctuated text often helps in boosting the performance of several downstream natural language understanding (NLU) tasks.

There is a plethora of work done in punctuation prediction over the past few decades. While some early methods of punctuation prediction used finite state or hidden markov models [1, 2], some other techniques have investigated probabilistic models like language modeling [3, 4, 5], conditional random fields (CRFs) [6, 7] and maximum entropy models [8]. As neural networks gained popularity, several approaches have been proposed based on sequence labeling and neural machine translation [9]. These models widely used convolutional neural networks (CNNs) and LSTM based architectures [10]. More recently, attention [11, 12] and transformer [13, 14, 15] based architectures which have been successfully applied to a wide variety of tasks, have shown to perform well for punctuation prediction.

Although it is a well explored problem in the literature, most of these improvements do not directly translate to all domains. In particular, punctuation prediction for conversational

speech is not very well explored [16, 17, 15]. Also, a number of approaches have been proposed exploiting the use of acoustic features in addition to lexical features for punctuation task, but they are rather limited and do not clearly address the gap in performance with ASR outputs. In this paper, we focus on multimodal semi-supervised deep learning approach for punctuation prediction in conversational speech by leveraging pre-trained lexical and acoustic encoders.

### 1.1. Relation to prior multimodal work

While several methodologies used either text or acoustic only information [18] for predicting punctuation, many studies show that combining both the features yields the best performance [13, 19, 20, 21]. Acoustic features widely used in the literature include prosodic information such as pause duration, phone duration, and pitch related values like fundamental frequency, and energy. [20] shows that using acoustic information lead to increased recognition of full stops. In [21], a hierarchical encoder is used to encode per frame acoustic features to word level features and the results show that incorporating acoustic features significantly outperform purely lexical systems. However, when trained on a very large independent text corpus, the lexical system outperformed the multimodal system that was trained on parallel audio/text corpora. To mitigate this, the work in [13] introduced speech2vec embeddings but they do not vary with respect to the acoustic context in reference speech.

In general, we identify two potential shortcomings with aforementioned multimodal systems. First, the training is still suboptimal due to lack of large-scale parallel audio/text corpora. Secondly, the models trained on reference text transcripts do not perform that well on ASR outputs, although incorporating acoustic features reduced the gap to some extent.

### 1.2. Novelty of this work

In this work, we introduce a novel framework for multimodal fusion of lexical and acoustic embeddings for punctuation prediction in conversational speech. Specifically, we investigate the benefits of using lexical and acoustic encoders that are pre-trained on large amounts of unpaired text and audio data using unsupervised learning. The key idea is to learn contextual representations through unsupervised training where substantial amounts of unlabeled data is available and then improve the performance on a downstream task like punctuation, for which the amount of data is limited, by leveraging learned representations. For multimodal fusion, we explore attention mechanism to automatically learn the alignment of word level lexical features and frame level acoustic features in the absence of explicit forced alignments.

We also show the adaptation of our proposed multimodal architecture for streaming usecase by limiting the future context. We further study the effect of pretrained encoders with re-

spect to varying data sizes and their performance when trained on very small amounts of data. Finally, we exploit the N-best lists from ASR to perform data augmentation and reduce the gap in performance when tested on ASR outputs.

## 2. Semi-supervised learning architecture

This section introduces our proposed *multimodal semi-supervised learning architecture* (MuSe) for punctuation prediction. We pose the prediction task as a sequence labeling problem where the model outputs a sequence of punctuation labels given text and corresponding audio. The architecture contains three main components: acoustic encoder, lexical encoder, and a fusion block to combine outputs from both the encoders. Figure 1 shows a schematic overview of the proposed approach.

The lexical encoder is pretrained on a large unlabelled text corpus for learning rich contextual representations and finetuned for the downstream task (i.e., weights are updated during punctuation model training). Given a sequence of input words  $(x_1^l, x_2^l, \dots, x_m^l)$ , subwords  $(s_1^l, s_2^l, \dots, s_n^l)$  are extracted using a wordpiece tokenizer [22]. The resulting subwords are fed as input to a pretrained encoder, which outputs a sequence of lexical features:  $H^l = (h_1^l, h_2^l, \dots, h_n^l)$  at its final layer.

The acoustic encoder takes audio signal as an input and outputs a sequence of frame level acoustic embeddings  $(x_1^a, x_2^a, \dots, x_T^a)$ . The acoustic encoder is pretrained on a large unlabelled audio data with the objective to predict future samples from a given signal context. This unsupervised pretraining is based on the work of Schneider et al. [23]. After pretraining, we freeze the parameters of the acoustic encoder. The frame level acoustic embeddings are then passed through a convolution layer followed by a uni-directional LSTM layer to learn task specific embeddings:  $\tilde{H}^a = (\tilde{h}_1^a, \tilde{h}_2^a, \dots, \tilde{h}_T^a)$ .

Since lexical and acoustic features differ in sequence length, it is not straightforward to concatenate them. Section 3 discusses about two different approaches for aligning acoustic feature sequence with lexical sequence. Once we obtain the resulting aligned acoustic sequence  $H^a = (h_1^a, h_2^a, \dots, h_n^a)$ , we concatenate last layer representations of pretrained lexical encoder ( $H^l$ ) with outputs from acoustic encoder ( $H^a$ ) and input to a linear layer with softmax activation to classify over the punctuation labels generating  $(\hat{p}_1, \hat{p}_2, \dots, \hat{p}_n)$  as outputs.

$$\hat{p}_i = \text{softmax}(W^k(h_i^l \oplus h_i^a) + b^k) \quad (1)$$

where  $W^k$ ,  $b^k$  denote weights and bias of linear output layer. The model is finetuned end-to-end to minimize the cross-entropy loss between the predicted distribution  $(\hat{p}_i)$  and targets  $(p_i)$ .

## 3. Multimodal fusion alignment

Several approaches have been proposed in the past for fusion of acoustic features with lexical encoder. Most of these approaches used word-level prosodic inputs and concatenated with lexical inputs or outputs from lexical encoder. In this section, we describe how we model frame-level acoustic features for fusion with sub-word lexical encoder using two different approaches: using force-aligned word durations and sub-word attention model.

### 3.1. Forced alignment fusion

Some prosodic features like fundamental frequency and energy can be averaged across each word and used as input to the acoustic encoder. Similarly, word duration can also be used as

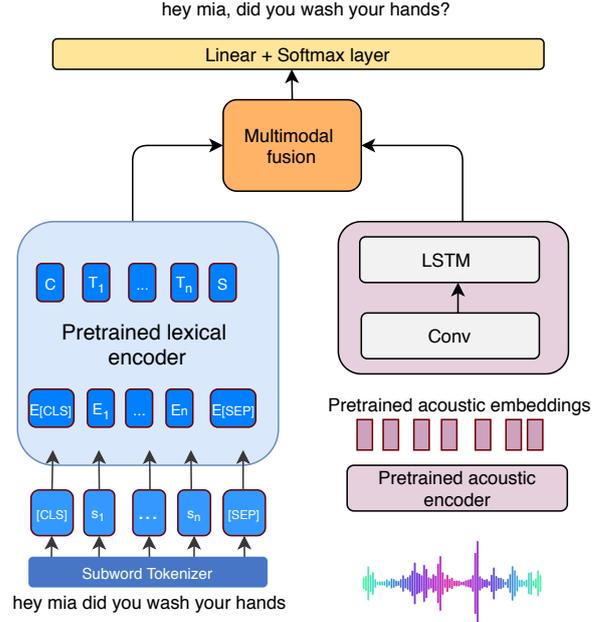


Figure 1: An overview of our multimodal semi-supervised learning architecture for punctuation prediction.

a feature and the work by [16] has shown minor improvements in punctuation prediction for conversational speech by employing relative word timing and duration of word. However, such mechanism does not capture the acoustic context beyond a word and also prevents the use of frame-level acoustic features where the average vector does not represent anything.

For this reason, we model frame-level acoustic features using an LSTM-based acoustic encoder where force-aligned word durations are used to obtain final word boundaries. We then use word boundaries to select the respective LSTM state outputs to form word-level features. We duplicate the same output to all the sub-words within each word.

### 3.2. Attention fusion

In the previous approach, we require force-aligned durations during training and word-level timestamps at inference time to form sub-word acoustic embeddings. While this is possible to achieve with conventional hybrid ASR systems, it may become an overhead when used in conjunction with end-to-end ASR systems<sup>1</sup> such as LAS and Transformer [24, 25]. The use of force-aligned durations may limit the acoustic context to a limited number of frames. For this reason, we introduce an attention module that uses scaled dot-product attention [26] to find the alignment between acoustic feature sequence with sub-word lexical sequence, operating on query  $Q$ , key  $\kappa$ , and value  $\vartheta$ :

$$\text{Attention}(Q, \kappa, \vartheta) = \text{softmax}\left(\frac{Q\kappa^T}{\sqrt{d_\kappa}}\right)\vartheta \quad (2)$$

where  $d_\kappa$  is the dimension of the keys. For attention model, we use the same encoder architecture as shown in Figure 1. However, since attention may not require such a low-resolution input acoustic sequence, we downsample the input feature sequence by using a fixed stride of 2 in the 1-dimensional convolution layer. In the attention module, key  $\kappa$  and value  $\vartheta$  are obtained

<sup>1</sup>Only in cases where punctuation is not modelled along with phonetic unit

from LSTM state outputs. Sub-word encoder outputs are used as query for this attention. The key  $\kappa$  is obtained by using a projection layer whose weight matrix is  $W^\kappa$ :

$$\kappa_i^a = f(W^\kappa, h_i^a) \quad (3)$$

The attention mechanism computes the attention weight according to the similarity between the query  $h_i^l$  and each key  $\kappa_i^a$ , and weighted sum of the values is then obtained using the attention weight.

$$h_i^{\tilde{a}} = \text{Attention}(h_i^l, \kappa_i^a, h_i^a) \quad (4)$$

The resulting aligned acoustic hidden vector is concatenated with lexical encoder output and given as input to the softmax function as explained in Section 2.

## 4. Experiments

### 4.1. Data

We conduct our experiments on English Fisher corpus [27]. The training data consists of 348 hours of conversational telephone speech where as the dev and test sets each consists of around 42 hours. To prepare the data splits, we took a subset of the full Fisher corpus to only include segments of a minimum length of six words in our data sets. Punctuation classes in the Fisher corpus are highly unbalanced (see Table 1), which is typical for conversational speech. Fisher corpus has separate time-annotated and punctuated transcripts. For the forced alignment fusion experiments, we need to compute the word boundary information from time-annotated transcripts using a pretrained acoustic model. For this purpose, we trained a TDNN-LSTM acoustic model with lattice-free Maximum Mutual Information (MMI) criterion [27] using the Kaldi ASR toolkit [28] and the same model is used for obtaining ASR transcriptions on test data. We restored punctuation marks like periods, commas, and question marks to the time-annotated transcripts by aligning them with the corresponding punctuated transcript.

Table 1: *Distribution of punctuation classes in Fisher corpus.*

Class	Count	Percentage
No punctuation	2,962,489	80.58
Comma (,)	70,927	11.83
FullStop (.)	362,166	6.26
Question mark (?)	56,128	1.33

### 4.2. Acoustic features

In addition to pretrained wav2vec features, we also experimented with two other prosodic features: pitch and melspec. The prosodic features are computed using a 25ms frame window with 10ms frame shift. We extracted F0 features based on Kaldi pitch tracker method [29], a highly modified version of the getf0 (RAPT) algorithm using Kaldi ASR toolkit [28]. Each frame is represented by 4-dimensional features consisting of - probability of voicing (pov) i.e the warped Normalized Cross Correlation Function(NCFF), normalized log pitch (the log-pitch with pov-weighted mean subtraction over 1.5 second window), delta pitch (time derivative of log-pitch) and raw log pitch. We also use a 80-dimension mel-scale spectrograms as alternative to pitch features as they have shown to transfer prosody well in text-to-speech systems [30].

For unsupervised wav2vec feature extraction, we train a wav2vec-large model<sup>2</sup> [23] on a 348-hour fisher audio corpus.

<sup>2</sup><https://github.com/pytorch/fairseq/blob/master/examples/wav2vec>

Table 2: *F1 scores for punctuation prediction using various acoustic features and two different fusion techniques; NP: No punctuation; FS: Fullstop; QM: Question mark;*

Model	Fusion	Feat	NP	Comma	FS	QM
BLSTM	-	-	96.2	69.4	66.1	74.0
BERT	-	-	96.5	71.3	71.1	78.4
MuSe	FA	pitch	97.3	74.1	74.6	80.4
		melspec	97.4	74.2	74.6	80.5
		wav2vec	97.5	75.6	75.6	81.3
MuSe	Att	pitch	97.3	73.5	73.4	79.0
		melspec	97.4	73.5	73.4	80.1
		wav2vec	97.5	75.5	73.4	81.3

For training, we preprocess the audio files by splitting each file into separate files of 10 to 30 seconds in length. The model is trained with the objective of constrastive loss and had 12 convolutional layers with skip connections. The output is a 512-dimensional unsupervised wav2vec representation.

### 4.3. Model Configurations

Our primary baseline model is a 4-layer BLSTM based on the work of Zelasko et. al. [16] with each layer having 128 weights in each direction. We also train another lexical only model which is a pretrained truncated BERT model [31] consisting of 6 transformer self attention layers with each hidden layer of size 768. The proposed (*MuSe*) model consists of a lexical encoder which is a pretrained truncated BERT model. The acoustic encoder used for learning task specific embeddings consists a convolutional layer of kernel size 5 and an LSTM hidden layer of size 256. We use a learning rate of 0.00002 and a dropout of 0.1 for the truncated BERT and MuSe models. For all experiments with pretrained lexical encoder, we use a subword vocabulary size of 28k<sup>3</sup>.

## 5. Results

### 5.1. Results using Multimodal framework

First, we compare the performance of our proposed multimodal architecture (*MuSe*) with baseline BLSTM model and lexical only BERT model (see Table 2). As expected, pure lexical BERT model outperformed BLSTM in all punctuation marks. We notice significant improvements (5%, and 4%) in Fullstop and Question Mark under F1 metric. This indicates that fine-tuning a pretrained lexical encoder for punctuation task outperforms the recurrent models that are trained from scratch and is synonymous with several other downstream tasks that are fine-tuned with BERT [31, 32, 33].

We now compare lexical only models to multimodal fusion models trained on three different features: pitch, melspec and wav2vec. Overall, we observe that using any kind of acoustic information helped in improving punctuation prediction across all three classes (Fullstop, Comma and Question Mark). This denotes that the fusion of acoustic features is still beneficial in conjunction with state-of-the-art pretrained lexical encoders as they model different aspects of punctuation.

Among acoustic features, pitch and melspec have shown similar performance improvements, except in Question mark when attention is used for fusion. This is understandable given that both pitch and melspec feature are extracted from audio using signal processing techniques and have been used as prosodic

<sup>3</sup>Although lexical encoder could be further pretrained on Fisher data, we didn't investigate it in this paper.

futures in the past [30]. Unsupervised wav2vec features proved to be the best among all acoustic features for multimodal fusion and its performance is significantly better ( $p < 0.01$ ) than semi-supervised lexical only BERT model with an absolute improvement of 4% on Comma, 2% on Fullstop and 3% on Question Mark. Comparing fusion techniques, we observe that forced alignment (*FA*) fusion performs slightly better than attention (*Att*) based fusion for Fullstop while the performance is similar on Comma and Question Mark. We hypothesize that this is because providing explicit acoustic information through duration labels helps better prediction of full stop as opposed to implicit learning through attention mechanism. Although our results are not directly comparable with the results provided in [16] on Fisher corpus (as the splits are different), we achieved better performance in all classes of punctuation.

## 5.2. Streaming models

We have conducted experiments to study the real time (streaming) performance of the proposed model when there is no future context. For the purpose of this experiment, we perform upper triangle masking (similar to transformer [26] decoder layers) in self attention layers of lexical encoder to mask the right side context. Since the acoustic encoder is unidirectional, we did not make any further changes. For the experimental results presented in Table 3, we have used forced alignment as fusion technique and wav2vec as acoustic features. We also trained an additional 4 layer LSTM model for baseline comparison. Similar to bidirectional models, the pretrained BERT model performs  $\sim 2\%$  better than LSTM model thus proving that pretraining also helps in learning better representations in the absence of right side context. We also observe that adding acoustic information leads to an additional  $\sim 3\%$  improvement over the lexical BERT model, confirming the effectiveness of our proposed approach for streaming usecase.

Table 3: *F1 scores for punctuation prediction (streaming).*

Model	NP	Comma	FS	QM
LSTM	95.4	67.6	68.7	72.3
BERT	95.6	68.5	69.7	75.1
MuSe	96.3	72.1	72.3	78.6

## 5.3. Ablation study: data sizes

We perform experiments on varying data sizes to study the effectiveness of our proposed approach. We experimented with data sizes of 1 hour, 10 hours and 100 hours and the results are reported in Table 4. For this study, we compared lexical only models (BLSTM and BERT) with our best performing model (MuSe) from Table 2. As expected, the performance of all three approaches were improved with the increase in amount of data size for both reference and ASR outputs. However, our proposed model (*MuSe*) fared significantly better than other two lexical models when trained on smaller datasets (1 hour and 10 hour). For comparison, our model achieved  $\sim 4-18\%$  absolute improvement in F1 score when trained on 1 hour of audio and text corpus.

Both pretrained models (lexical BERT and *MuSe*) performed very well on reference transcripts when compared with BLSTM models. However, the gap was significantly reduced when tested on ASR outputs. Our proposed model (*MuSe*) performed better than lexical BERT due to the fusion of acoustic features but the performance gap on ASR outputs is still quite evident. This shows that although pretrained models perform

Table 4: *F1 scores for punctuation on varying data sizes.*

Model	hours	Comma		FS		QM	
		Ref	ASR	Ref	ASR	Ref	ASR
BLSTM	1	49.7	49.8	43.9	44.3	29.7	28.0
BERT		54.5	54.0	51.2	50.8	42.5	35.7
MuSe		58.8	57.4	55.2	55.2	48.3	40.1
BLSTM	10	60.9	58.1	53.5	51.9	46.3	42.4
BERT		65.6	62.0	62.5	58.9	61.6	57.8
MuSe		68.2	63.5	65.9	62.9	72.9	60.8
BLSTM	100	68.9	62.8	66.1	63.4	72.8	65.6
BERT		70.1	65.5	69.3	64.8	75.8	68.6
MuSe		71.6	66.2	70.8	65.7	77.1	69.5

well on reference transcripts with smaller datasets, they are not yet robust to ASR errors. This is due to the fact that pretrained masked language models like BERT were trained only on reference transcripts and have not seen the grammatical errors that are introduced by ASR.

## 5.4. Robustness to ASR errors

We have seen that models trained on reference transcripts did not perform that well when tested on ASR outputs. To make models more robust against ASR errors, we perform data augmentation with ASR outputs for training [15]. For punctuation restoration, we use edit distance measure to align ASR hypothesis with reference punctuated text and restore the punctuation from each word in reference transcription to hypothesis. If there are words that are punctuated in reference but got deleted in ASR hypothesis, we restore the punctuation to previous word. We performed experiments with data augmentation using N-best lists and the results are reported in Table 5.

From the results, it is evident that the models trained purely on reference transcripts were outperformed by models trained on augmented text (both reference and ASR outputs). The last three rows from Table 5 indicate that the data augmentation approach yielded better performance in all classes of punctuation. Overall, data augmentation with 3-best lists gave the best performance. Question mark improved by 6% in F1 score by performing data augmentation. The improvement might be due to increased number of training examples in augmented data.

Table 5: *Comparison of F1 scores for punctuation with models trained on reference transcripts and ASR augmented data.*

Model	n-best	NP	Comma	FS	QM
BLSTM-Ref	-	94.5	63.5	63.8	66.7
BERT-Ref	-	95.2	65.5	64.1	68.3
MuSe-Ref	-	95.6	67.2	66.7	70.6
MuSe-ASR	1-best	95.8	68.5	69.0	75.7
	3-best	95.6	69.0	69.5	76.4
	5-best	95.5	67.3	66	76.3

## 6. Conclusions

We introduced a novel multimodal semi-supervised learning framework which leverages large amounts of unlabelled audio and text data for punctuation prediction. We proposed an alternative attention based multimodal fusion mechanism which is effective, in the absence of forced alignment word durations. Through our data sizes ablation study, we showed how our proposed model is superior in performance to lexical only models on reference transcripts. In order to address the performance gaps on ASR outputs, we presented a robust model that is less affected by ASR errors by performing data augmentation with N-best lists.

## 7. References

- [1] Y. Gotoh and S. Renals, "Sentence boundary detection in broadcast speech transcripts," 2000.
- [2] H. Christensen, Y. Gotoh, and S. Renals, "Punctuation annotation using statistical prosody models," in *ISCA tutorial and research workshop (ITRW) on prosody in speech recognition and understanding*, 2001.
- [3] A. Stolcke, E. Shriberg, R. Bates, M. Ostendorf, D. Hakkani, M. Plauche, G. Tur, and Y. Lu, "Automatic detection of sentence boundaries and disfluencies based on recognized words," in *Fifth International Conference on Spoken Language Processing*, 1998.
- [4] D. Beeferman, A. Berger, and J. Lafferty, "Cyberpunc: A lightweight punctuation annotation system for speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2. IEEE, 1998, pp. 689–692.
- [5] A. Gravano, M. Jansche, and M. Bacchiani, "Restoring punctuation and capitalization in transcribed speech," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 4741–4744.
- [6] W. Lu and H. T. Ng, "Better punctuation prediction with dynamic conditional random fields," in *Proceedings of the 2010 conference on empirical methods in natural language processing*, 2010, pp. 177–186.
- [7] N. Ueffing, M. Bisani, and P. Vozila, "Improved models for automatic punctuation prediction for spoken and written text." in *Interspeech*, 2013, pp. 3097–3101.
- [8] J. Huang and G. Zweig, "Maximum entropy model for punctuation annotation from speech," in *Seventh International Conference on Spoken Language Processing*, 2002.
- [9] A. Öktem, M. Farrús, and L. Wanner, "Attentional parallel rnns for generating punctuation in transcribed speech," in *International Conference on Statistical Language and Speech Processing*. Springer, 2017, pp. 131–142.
- [10] V. Pahuja, A. Laha, S. Mirkin, V. Raykar, L. Kotlerman, and G. Lev, "Joint learning of correlated sequence labelling tasks using bidirectional recurrent neural networks," *arXiv preprint arXiv:1703.04650*, 2017.
- [11] O. Tilk and T. Alumäe, "Lstm for punctuation restoration in speech transcripts," in *Sixteenth annual conference of the international speech communication association*, 2015.
- [12] O. Tilk and T. Alumäe, "Bidirectional recurrent neural network with attention mechanism for punctuation restoration." in *Interspeech*, 2016, pp. 3047–3051.
- [13] J. Yi and J. Tao, "Self-attention based model for punctuation prediction using word and speech embeddings," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7270–7274.
- [14] B. Nguyen, V. B. H. Nguyen, H. Nguyen, P. N. Phuong, T.-L. Nguyen, Q. T. Do, and L. C. Mai, "Fast and accurate capitalization and punctuation for automatic speech recognition using transformer and chunk merging," *arXiv preprint arXiv:1908.02404*, 2019.
- [15] M. Sunkara, S. Ronanki, K. Dixit, S. Bodapati, and K. Kirchhoff, "Robust prediction of punctuation and truecasing for medical asr," in *Proceedings of the First Workshop on Natural Language Processing for Medical Conversations*, 2020, pp. 53–62.
- [16] P. Żelasko, P. Szymański, J. Mizgajski, A. Szymczak, Y. Carmiel, and N. Dehak, "Punctuation prediction model for conversational speech," *arXiv preprint arXiv:1807.00543*, 2018.
- [17] Ł. Augustyniak, P. Szymanski, M. Morzy, P. Zelasko, A. Szymczak, J. Mizgajski, Y. Carmiel, and N. Dehak, "Punctuation prediction in spontaneous conversations: Can we mitigate asr errors with retrofitted word embeddings?" *arXiv preprint arXiv:2004.05985*, 2020.
- [18] A. Moró and G. Szaszák, "A prosody inspired rnn approach for punctuation of machine produced speech transcripts to improve human readability," in *2017 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*. IEEE, 2017, pp. 000 219–000 224.
- [19] O. Klejch, P. Bell, and S. Renals, "Punctuated transcription of multi-genre broadcasts using acoustic and lexical approaches," in *IEEE Spoken Language Technology Workshop (SLT)*, 2016, pp. 433–440.
- [20] H. Christensen, Y. Gotoh, and S. Renals, "Punctuation annotation using statistical prosody models," in *ISCA tutorial and research workshop (ITRW) on prosody in speech recognition and understanding*, 2001.
- [21] O. Klejch, P. Bell, and S. Renals, "Sequence-to-sequence models for punctuated transcription combining lexical and acoustic features," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5700–5704.
- [22] M. Schuster and K. Nakajima, "Japanese and korean voice search," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 5149–5152.
- [23] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.
- [24] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.
- [25] L. Dong, S. Xu, and B. Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.
- [26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [27] C. Cieri, D. Miller, and K. Walker, "The fisher corpus: a resource for the next generations of speech-to-text." in *LREC*, vol. 4, 2004, pp. 69–71.
- [28] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.
- [29] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2014, pp. 2494–2498.
- [30] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," *arXiv preprint arXiv:1803.09047*, 2018.
- [31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [32] Q. Chen, Z. Zhuo, and W. Wang, "Bert for joint intent classification and slot filling," *arXiv preprint arXiv:1902.10909*, 2019.
- [33] W. Yang, Y. Xie, A. Lin, X. Li, L. Tan, K. Xiong, M. Li, and J. Lin, "End-to-end open-domain question answering with bertserini," *arXiv preprint arXiv:1902.01718*, 2019.