IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 October 2022, accepted 25 October 2022, date of publication 4 November 2022, date of current version 9 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3219838

RESEARCH ARTICLE

Automatic Speech Recognition Post-Processing for Readability: Task, Dataset and a Two-Stage Pre-Trained Approach

JUNWEI LIAO[1], YU SHI[2], AND YONG XU[3]
[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]Microsoft Cognitive Services Research Group, Redmond, WA 98052, USA
[3]Westone Information Industry Inc., Chengdu 610095, China

Corresponding author: Junwei Liao (liaojunwei@std.uestc.edu.cn)

ABSTRACT Nowadays Automatic Speech Recognition (ASR) systems can accurately recognize which words are said. However, due to the disfluency, grammatical error, and other phenomena in spontaneous speech, the verbatim transcription of ASR impairs its readability, which is crucial for human comprehension and downstream tasks processing that need to understand the meaning and purpose of what is spoken. In this work, we formulate the ASR post-processing for readability (APR) as a sequence-to-sequence text generation problem that aims to transform the incorrect and noisy ASR output into readable text for humans and downstream tasks. We leverage the Metadata Extraction (MDE) corpus to construct a task-specific dataset for our study. To solve the problem of too little training data, we propose a novel data augmentation method that synthesizes large-scale training data from the grammatical error correction dataset. We propose a model based on the pre-trained language model to perform the APR task and train the model with a two-stage training strategy to better exploit the augmented data. On the constructed test set, our approach outperforms the best baseline system by a large margin of 17.53 on BLEU and 13.26 on readability-aware WER (RA-WER). The human evaluation also shows that our model can generate more human-readable transcripts than the baseline method.

INDEX TERMS Automatic post-editing, ASR post-processing for readability, data augmentation, pre-trained language model, natural language processing.

I. INTRODUCTION

ASR systems have reached great recognition accuracy, even outperforming professional human transcribers on conversational telephone speech in terms of Word Error Rate (WER), thanks to the fast advancement of speech-to-text technology [1]. However, spontaneous speech is riddled with disfluency, informal expression, grammatical errors, and other noises that make it difficult to comprehend. For example, in an utterance such as "I want a flight ticket to Boston, uh, I mean to Denver on Friday", the speaker means to communicate "I want a flight ticket to Denver on Friday." The segment "to Boston, uh, I mean" in the speech transcript is not helpful for interpreting the intent of the sentence. While the ASR system is excellent at distinguishing which words are said, it tends to transcribe the speech verbatim while ignoring the readability of the output text. Therefore, for the above example, ASR systems optimized for recognition accuracy will keep all words including disfluencies, increasing the cognitive load of the reader and impairing the performance of downstream tasks.

Automatic speech transcription that is highly readable for humans is required for applications like automatic subtitle generation [2], [3] and meeting minutes generation [4], [5], while machine translation [6], [7], dialogue systems [8], [9], voice search [10], [11], voice question answering [12], [13], and many other applications require highly readable transcriptions to generate the best machine response. If the system

The associate editor coordinating the review of this manuscript and approving it for publication was Aysegul Ucar.

is unable to deal with deficiencies in speech transcription, it will have a substantial negative impact on the application users' experience.

In this paper, we formulate generating human-readable text from ASR transcripts as a sequence-to-sequence (seq2seq) text generation problem, which we call the ASR post-processing for readability (APR). The APR aims to transform the ASR output into a readable text for humans and downstream natural language processing (NLP) tasks. Readability in this context means having proper segmentation, capitalization, and no grammatical errors or speech disfluencies. The APR can be treated as a style transfer, converting informal speech to formal written language.

Since there is no off-the-shelf dataset for the APR task, we construct the desired dataset from the RT-03 MDE Training Data [14], which includes speech audio, human transcript, and annotation. We follow the annotation guideline [15] to parse the human transcript and annotation file to get the speech and readable transcript pairs. After data processing, we obtained about 27k samples for the APR task. To solve the training data scarcity problem, in addition to directly fine-tuning the model, we propose a novel data augmentation method that synthesizes large-scale training data using Grammatical Error Correction (GEC) corpora. First, we used a text-to-speech (TTS) system to convert the ungrammatical sentences to speech. Then, we use an ASR system to transcribe the TTS output. Finally, we use the output of the ASR system and the original grammatical sentences to create the data pairs. By this means, we produced 1.1 million training samples for the APR task.

We build a Transformer-based model to perform the APR task. Our model is based on the Robustly Optimized BERT Pretraining Approach (RoBERTa) [16], which is a pre-trained language model used for natural language understanding tasks. Inspired by the UNIfied pre-trained Language Model (UniLM) [17], which applies self-attention masks on the Bidirectional Encoder Representations from Transformers (BERT) [18] to convert it into a seq2seq model, we adapt the RoBERTa towards a seq2seq model for the APR task with a specific self-attention mask and autoregressive prediction. We use a two-stage training strategy, consisting of pre-training and fine-tuning, in order to maximize the benefit of the augmented data. Because of this two-phase training, not only is it possible to get important knowledge from the augmented data, but it also eliminates the possibility of being overwhelmed and negatively influenced by augmented data.

On the test split of the constructed dataset, we evaluate our approach and compare it with multiple baseline systems. Our approach beats the best baseline system that includes an extra step of removing disfluencies by 17.53 on BLEU and 13.26 on readability-aware WER (RA-WER), respectively. The results of the human evaluation also reveal that our approach produces more human-readable transcripts for ASR output than the baseline method.

Our main contributions can be summarized as follows:

- We formulate the problem of making ASR output more human-readable to a seq2seq text generation problem: ASR post-processing for readability (APR). It aims to solve the shortcomings of the traditional post-processing concept/methods by jointly performing error correction and readability improvements in one step.
- We construct a dataset for training and evaluating the APR task. To address the lack of training data, we propose a novel data augmentation method to synthesize large-scale training data to pre-train the language model.
- Utilizing a two-stage training strategy, our proposed model outperforms the baselines in both automatic and human evaluation.

We note that a shorter conference version of this paper appeared in Liao et al. [19]. Our initial conference paper omitted many details due to the page limit and only compared with an in-house baseline system in the experiment. This manuscript complements the missing information on datasets and approaches, and provides extensive experiments and further discussions. The additional experiments include:

- We compare the proposed approach with several online speech-to-text (STT) services apart from an in-house STT system.
- We analyze the contribution of each step of the baseline system to the final readability with respect to reference sentences.
- We make a comparison of using other models instead of our proposed model to perform the APR task.
- Besides the ASR system we use in this work, we also conduct an experiment to see how much gain the proposed approach provides on top of different ASR systems.

II. RELATED WORK

A. ASR POST-PROCESSING

ASR post-processing is the process of editing the output transcripts of an ASR system, where the ASR system is usually used as a black box and users do not have access to the internals of the system. The purpose of ASR post-processing can be divided into two categories: improving the performance of the ASR system in terms of recognition accuracy or meeting the final goal of the system.

One type of ASR post-processing is to enhance the performance of ASR systems. The optimization goal of ASR systems is to pursue high recognition accuracy. Post-processing can further improve this metric without changing the ASR system. [20] contains an overview of previous works on error detection and correction for ASR. Besides, Guo et al. [21] trained an LSTM-based seq2seq model to correct spelling errors. Hrinchuk et al. [22] investigated the use of Transfomer-based architectures for the correction of ASR output into grammatically and semantically correct forms.

The other type of ASR post-processing is to change the form of ASR output to meet the requirements of human understanding or downstream task handling. This type of

processing usually modifies the sentence display format, such as adding capitalization and punctuation [23], [24], [25], [26], [27], [28], correcting grammatical errors [29], [30], [31], [32], removing disfluencies [33], [34], and formatting dates, times, and other numerical entities [35].

Unlike the two aforementioned types of ASR post-processing, our APR task post-process the ASR output to obtain the highly readable text for human and downstream tasks. Readability in this context means having proper segmentation, capitalization, and no grammatical errors or speech disfluencies. From this definition, we can see that our APR task contains both types of ASR post-processing mentioned above.

B. METADATA EXTRACTION

Metadata extraction (MDE) [36] shares the goal of our APR task, which is making ASR transcripts more human-readable. The MDE research, which is part of the DARPA EARS program [37], intends to improve speech recognition output by adding automatically tagged information on the location of sentence boundaries, speech disfluencies, and other key phenomena. The purpose of MDE is to allow technologies that can improve raw Speech-To-Text output into forms that are more useful to people and downstream automated processes. Simply put, this entails the generation of automated transcripts that are as readable as possible. On top of verbatim transcription, MDE divides the aim into many classification tasks. The Hidden Markov Model [38], [39], Maximum Entropy, and Conditional Random Fields [38], [40] approaches are used in most MDE systems to handle both the textual and prosodic information.

There are three main differences between the APR and the MDE. Firstly, The MDE decomposes the task goal into multiple classification tasks to obtain the annotated text, which then requires subsequent processing to get the final result. While the APR directly gets the readable text from the raw text in an end-to-end manner. Secondly, the methods adopted by the MDE require the input of textual and prosodic information, while the APR only requires text information as the input. Thirdly, while the MDE improves the readability of ASR transcripts to a certain extent, it ignores the recognition errors introduced by the ASR system. Thus the MDE needs to work with other ASR post-processing components such as language model rescoring to provide the final human-readable transcript. The APR can simultaneously correct recognition errors from ASR systems without additional processing steps.

C. PRE-TRAINED LANGUAGE MODEL

Recently pre-training approaches [41], [42], [43] have been used for many NLP tasks. Large language models pre-trained on massive text collections have shown surprising emergent capabilities to generate text and perform zero- and few-shot learning [44], [45], [46], [47], [48]. The most successful pre-trained language models are based on the Transformer [49] architecture and trained with self-supervised tasks such as mask language model, and denoising autoencoders. Among them, BERT [18] and RoBERTa [16] are single-stack Transformer encoders; GPT & GPT-2 [50], [51] and XLNET [52] are single-stack Transformer decoders; UniLM [17] is a single-stack Transformer serving both encoder and decoder roles; and MASS [53], BART [54] and T5 [55] are standard Tranformer encoder-decoder architectures.

We adapt RoBERTa with the self-attentive mechanism to support the seq2seq objective, which preserves the excellent performance of RoBERTa on natural language understanding tasks while enabling it to be used for natural language generation tasks.

III. DATASETS

A. APR DATASET CONSTRUCTION

To the best of our knowledge, there is no readily available dataset that can be utilized for the APR task. As a result, we create the necessary dataset from the MDE corpus,[1] namely the MDE RT-03 Training Data Text and Annotations corpus. Specifically, this data was gathered to support the DARPA EARS (Efficient, Affordable, Reusable Speech-to-Text) Program in Metadata Extraction (MDE), which pursues an objective that is quite similar to the APR. The data in this release comprises transcripts and annotations from English Conversational Telephone Speech (CTS) and Broadcast News (BN). The conversational speech from CTS has more grammatical faults and speech disfluencies, making it more difficult for the model to convert into readable scripts, but it is also more prevalent in reality. As a result, we solely utilize CTS data to construct the dataset for the APR task.

The CTS data was drawn from the Switchboard corpus [56], of which the transcripts and annotations cover approximately 40 hours of CTS audios of casual and conversational speech. The annotation files are provided in RTTM[2] format developed by NIST to support the EARS Program. The RTTM format labels each token in the reference transcript according to the properties it displays, such as lexeme, non-lexeme, edit, filler, segment, etc. Next, we briefly describe these properties used in annotation.

Annotators identified four sorts of fillers in the MDE annotation standard [15]: *filled pauses* like ''uh'' and ''um'', *discourse markers* like ''you know'', *asides and parentheticals*, and *editing terms* like ''sorry'' and ''I mean''. *Edit disfluencies* are also recognized, with the full extent of the disfluency (or string of adjacent disfluencies) and *interruption points (IP)* annotated. Annotators also identify SUs (alternately semantic units, sense units, syntactic units, slash units, or sentence units), which are units within the discourse that serve to communicate a speaker's whole thinking or idea. The purpose of SU labeling, like that of disfluency annotation, is to increase transcript readability by presenting information in short, organized, coherent pieces rather than

[1] https://catalog.ldc.upenn.edu/LDC2004T12

[2] http://www.nist.gov/speech/tests/rt/rt2003/fall/index.htm

long turns or stories. There are four types of sentence-level SUs: *statements*, *questions*, *backchannels*, and *incomplete* SUs. The annotation process additionally determines a number of sub-sentence SU boundaries (*coordination* and *clausal* SUs) to improve inter-annotator consistency.

By parsing the annotation files, we get the transcript with metadata annotation, which, for example, uses '`/.`' for *statement boundaries (SU)*, '`<>`' for *fillers*, '`[]`' for *disfluency edit* words, and '`*`' for *interruption points* inside edit disfluencies. The following example shows an ASR transcript with metadata annotation:

```
and < uh > < you know > wash your clothes
wherever you are /. and [ you ] * you
really get used to the outdoors ./
```

The transcripts containing annotated metadata must be processed in order to produce a human-readable text. By cleaning up the metadata annotations, we construct a readable target transcript, in which the deletable section of edit disfluencies and fillers is deleted, and each SU is presented as a separate line inside the transcript. To enhance the readability of the transcript, we uppercase the first word of each sentence. After these processes, the above transcript with metadata annotation becomes a readable text: "And wash your clothes wherever you are. And you really get used to the outdoors."

After the above processing, we obtain 27,355 readable transcripts in total. The corresponding speech duration is about 34 hours. By pairing them with corresponding speech transcripts, we obtain the input and output samples for the proposed APR model (Section IV-B). About 1K samples are extracted for validation and testing, respectively. We ensure that samples of training, validation, and testing come from different conversations.

B. AUGMENTED DATA SYNTHESIS

Typically, transformer-based models are trained on millions of parallel sentence pairs, and they have a high tendency to overfit when the data is insufficient. To overcome this challenge, Hrinchuk et al. [22] proposed two self-complementary regularization strategies. Besides initializing the model weights using the pre-trained language model, the other solution is data augmentation. Inspired by this, we propose a novel data augmentation method for the APR.

By starting with a grammatical error correction (GEC) dataset as the seed data, we are able to generate large-scale training data. The GEC aims to correct different kinds of errors such as spelling, punctuation, grammatical, and word choice errors [57]. The GEC data samples contain grammatically correct and incorrect sentence pairs. A human corrects the grammatically incorrect sentence to obtain the target grammatically correct sentence. Table 1 shows an example sentence pair taken from the GEC dataset for illustration. With its help, we can incorporate more forms of mistakes that are not exclusive to our ASR system, allowing us to make the APR model more general.

TABLE 1. A GEC data sample is shown. The input is a sentence with some grammatical errors. The output is a grammatically correct sentence.

Input	*She see Tom is catched by policeman in park at last night.*
Output	*She saw Tom caught by a policeman in the park last night.*

Figure 1 depicts the steps involved in data synthesis. The ungrammatical sentences are first converted to speech using a text-to-speech (TTS) system. Then, these audio files are sent into an ASR system, which generates the corresponding transcripts. The generated text incorporates both the grammatical faults discovered originally in the GEC dataset as well as the problems discovered through the TTS plus ASR pipeline. In the last step, we pair the ASR system's outputs with the original grammatical sentences as the training examples for the APR task.

Specifically, the TTS system that we use is a Tacotron2 model [58], which is composed of a recurrent seq2seq feature prediction network that maps character embeddings to mel-scale spectrograms, followed by a modified WaveNet model acting as a vocoder to synthesize time-domain waveforms from those spectrograms. Trained directly on normalized character sequences and corresponding speech waveforms, Tacotron2 can synthesize natural sounding speech that is difficult to distinguish from real human speech. We fed the grammatically incorrect sentences from the seed corpus into Tacotron2 to produce the audio files simulating human speakers. To simulate the diversity of speech, we configure the TTS system using 220 speaker features to synthesize the speech audio. These speaker features diversify with different gender, speaking rates, prosody, etc, which imitate the various speakers in the real world. Each sentence randomly selected one speaker, and all speakers have the same number of input sentences [59]. In addition to the mentioned simulation method, we also tried using the top-k best output of the ASR system to augment our dataset tenfold. However, we found that the augmented dataset is not beneficial, due to the lack of diversity in the resulting sentences, which often differ only in some characters.

The ASR system that we use is a hybrid model using contextual layer trajectory LSTM (cltLSTM) [60] for acoustic modeling. It decouples the responsibilities of temporal modeling and target classification from each other, using time and depth LSTMs, respectively, and integrates future context frames in order to get additional information for accurate acoustic modeling. The input feature is an 80-dimension log Mel filter bank that is activated for every 20 milliseconds (ms) of speech, which is accomplished via the use of frame skipping [61]. The senone labels are modeled by 9404 nodes in the softmax layer. Runtime decoding is carried out utilizing a 5-gram LM with a decoding graph of around 5 gigabytes (Gbs). The cltLSTM has a lookahead of 24 frames, which

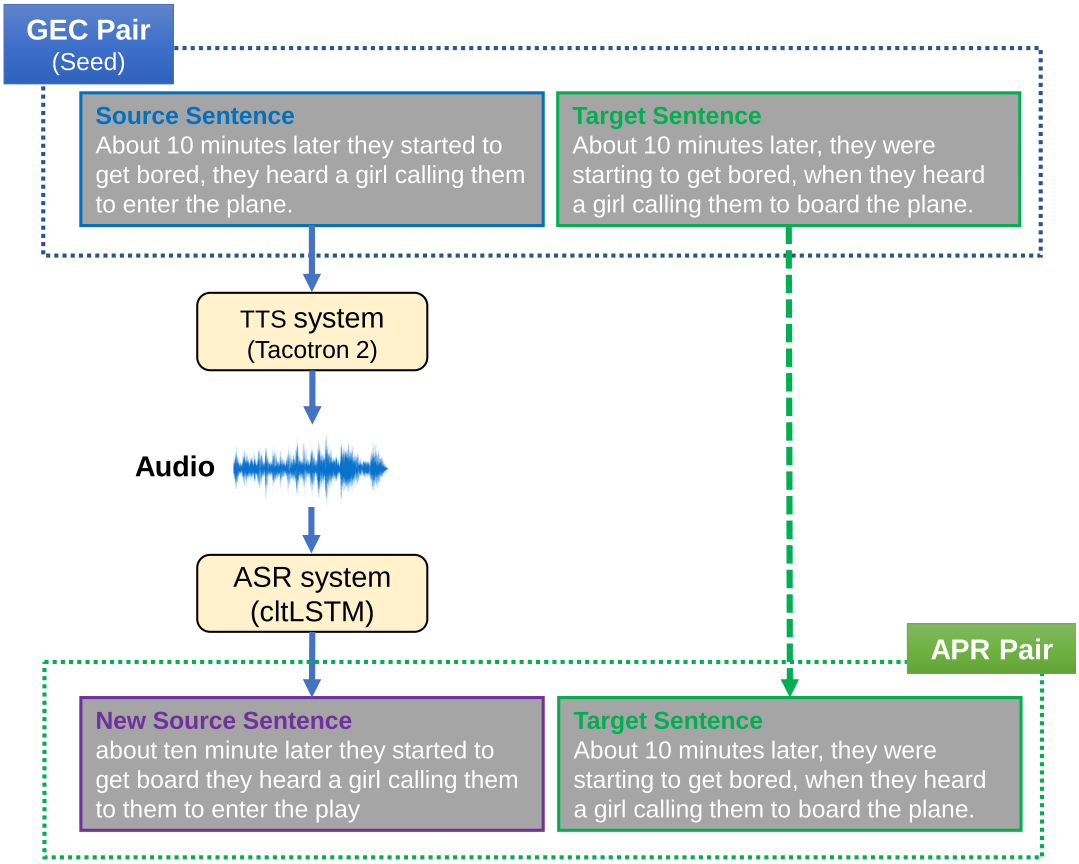

FIGURE 1. The process of data synthesis is shown. The top sentence pair is from the GEC dataset. The bottom sentence pair is an APR instance. The source sentence of the APR is obtained from the source sentence of the GEC dataset processed by the TTS and ASR systems. The target sentence of the GEC dataset remains unchanged and is used as the target sentence for the APR.

amounts to a period of 480 milliseconds. The training of the cltLSTM model exploits a three-stage training strategy: from cross-entropy to maximum mutual information [62], and then followed by sequential teacher-student learning [63].

We used the data from the datasets provided by restricted tracks of BEA 2019 shared task [64] as our seed corpora for data synthesis. Specifically, we collected data from FCE [65], Lang-8 Corpus of Learner English [66], [67], and W&I+LOCNESS [64], [68], totaling to around 1.1 million samples. The total duration of speech data synthesized using the TTS system is about 1335 hours. Through the above process, we obtained the augmented training dataset containing 1.1M sentence pairs. The statistics of the augmented dataset are shown in Table 2.

TABLE 2. Dataset statistics are shown. We create synthetic data from the seed corpus using the synthesis process described in Section III-B. Seed corpus FCE, W&I+LOCNESS, and Lang-8 Corpus are used to synthesize the augmented training data.

Seed corpus		Synthetic data
GEC dataset	Sent pairs	Sent pairs
FCE	28,350	1,100,219
W&I+LOCNESS	34,308	
Lang-8 Corpus	1,037,561	

IV. APPROACHES

In this section, we first introduce the multi-head attention network of the Transformer, through which we adapt the RoBERTa to a sequence-to-sequence (seq2seq) generative model (Section IV-A). Then, we describe the proposed model based on RoBERTa for the APR task (Section IV-B). Finally, we present a two-stage training strategy (Section IV-C) to better utilize the augmented data to train the proposed APR model.

A. BACKGROUND

As shown in the left part of Figure 2, the proposed APR model is based on RoBERTa [16], which is a robustly optimized BERT [18] pre-training approach. BERT and RoBERTa both contain a single Transformer stack. Transformer is a commonly used model architecture with multiple layers and each layer is composed of a multi-head attention network followed by a feed-forward neural network. The modification of our model to RoBERTa is mainly implemented through multi-head attention. Therefore, we introduce the multi-head attention network in Transformer.

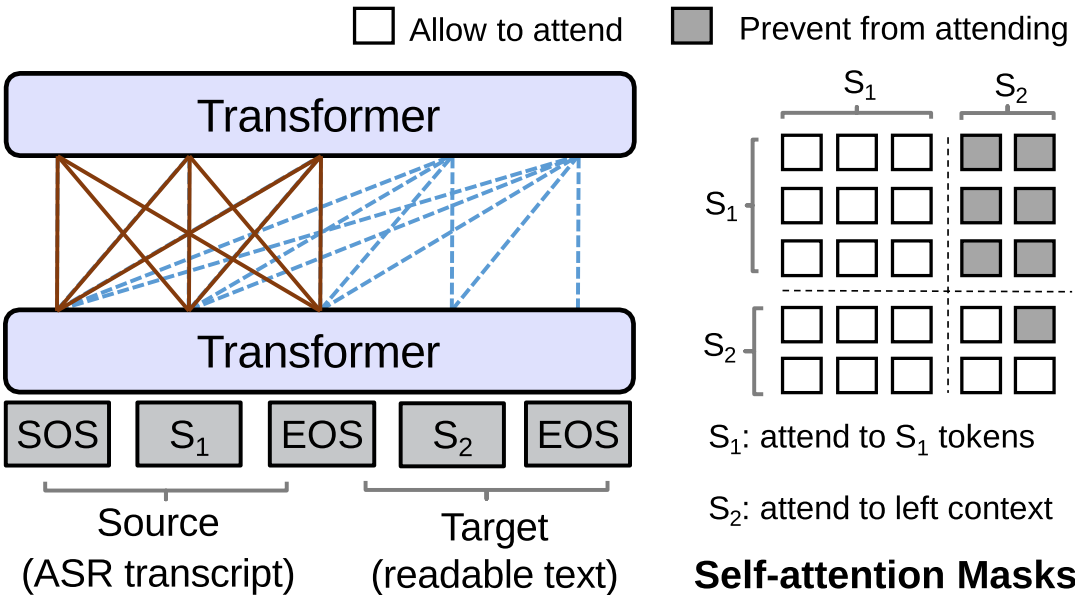

FIGURE 2. The schematic diagram of the proposed APR model. The left part is the proposed APR model based on the modified RoBERTa. The right part is the self-attention mask of the Transformer layer for the seq2seq task.

Specifically, the input sequence $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is first embedded into $\mathbf{H}^0 = [\mathbf{h}_1^0, \ldots, \mathbf{h}_n^0]$, and then encoded into contextual representations at different levels of abstract $\mathbf{H}^l = [\mathbf{h}_1^l, \ldots, \mathbf{h}_n^l]$ using an L-layer Transformer $\mathbf{H}^l = \text{Transformer}_l(\mathbf{H}^{l-1})$, where Transformer_l is the l-th Transformer layer and $l \in [1, L]$. To aggregate the output vectors of the previous layer, multi-head attention is adopted in each Transformer layer. The output of one head of the multi-head attention $\textbf{Attention}_l$ is computed for the l-th Transformer layer as follows:

$$\mathbf{Q}_l = \mathbf{H}^{l-1}\mathbf{W}_l^Q,\ \mathbf{K}_l = \mathbf{H}^{l-1}\mathbf{W}_l^K,\ \mathbf{V}_l = \mathbf{H}^{l-1}\mathbf{W}_l^V \tag{1}$$

$$\mathbf{M}_{ij} = \begin{cases} 0, & \text{allow to attend} \\ -\infty, & \text{prevent from attending} \end{cases} \tag{2}$$

$$\textbf{Attention}_l = \text{softmax}\left(\frac{\mathbf{Q}_l\mathbf{K}_l^\top}{\sqrt{d_k}} + \mathbf{M}\right)\mathbf{V}_l \tag{3}$$

where the previous layer's output $\mathbf{H}^{l-1} \in \mathbb{R}^{n \times d_h}$ is linearly projected to a triple of queries, keys and values using parameter matrices $\mathbf{W}_l^Q, \mathbf{W}_l^K, \mathbf{W}_l^V$ respectively, and the mask matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ decides whether a pair of tokens can be attended to each other. When computing a token's contextualized representation, the mask matrix $\mathbf{M}$ is utilized to determine what context it can attend to.

B. APR MODEL

RoBERTa is pre-trained exclusively via bidirectional LM, which allows all tokens to attend to each other in prediction. As indicated in Equation 2, the self-attention mask $\mathbf{M}$ is a zero matrix, so that every token is allowed to attend across all positions in the input sequence. This setting makes its behavior more discriminative rather than generative. In their speech recognition correction work, Hrinchuk et al. [22] showed the success of transfer learning from BERT to the seq2seq task by initializing both the encoder and the decoder with pre-trained BERT. Inspire by their work, we follow UniLM [17] and use specific self-attention masks to the RoBERTa model to select what context the prediction conditions on. In this case, it is transformed into a seq2seq generation model. During the training process, we use an autoregressive technique to achieve whole-sentence prediction rather than only masked-position predictions. Another advantage of using this strategy is that the model is capable of accurately predicting the end of a sentence. So there is no need to tweak the maximum output length and length penalty in the same way that the UniLM fine-tuning is accomplished.

Specifically, the mask matrix $\mathbf{M}$ used for the seq2seq objective is shown in the right part of Figure 2. The left part of $\mathbf{M}$ is set to 0 in order that all tokens are able to attend to the first segment. The top right part of $\mathbf{M}$ is set to $-\infty$ in order to prevent attention from the source segment to the target segment. Furthermore, we set the upper triangular section of the bottom right part of $\mathbf{M}$ to $-\infty$, while the other elements are set to 0, preventing tokens in the target segment from attending their future (right) positions.

After adapting to a generative model via the custom attention mask, the APR model is trained using seq2seq [6], [69] learning. Specifically, given a source sentence $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, a seq2seq model learns to generate its target sentence $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$. The model is usually trained by maximizing the log-likelihood of the training source-target sentence pairs:

$$\begin{aligned} L(\boldsymbol{\theta}; \mathcal{D}) &= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \log P\left(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta}\right) \\ &= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \log \prod_{t=1}^{m} P\left(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}; \boldsymbol{\theta}\right) \end{aligned} \tag{4}$$

where $\mathcal{D}$ denotes the training set (i.e., source-target parallel sentence pairs) and $\boldsymbol{\theta}$ denotes the parameters of the model. During inference, the decoder generates output $\boldsymbol{y}$ autoregressively by maximizing $P(\boldsymbol{y} \mid \boldsymbol{x}; \hat{\boldsymbol{\theta}})$:

$$P(\boldsymbol{y} \mid \boldsymbol{x}; \hat{\boldsymbol{\theta}}) = \prod_{t=1}^{m} P(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}; \hat{\boldsymbol{\theta}}) \tag{5}$$

where $\hat{\boldsymbol{\theta}}$ denotes the parameters of the model learned in training.

In the training phase, we concatenate the source and target segments with special tokens as the input and train the model with teacher-forced maximum likelihood. For example, given source segment $t_1t_2t_3$ and its target segment t_4t_5, we feed input "$[SOS]t_1t_2t_3[EOS]t_4t_5[EOS]$" into the model. In the inference phase, we feed the model "$[SOS]t_1t_2t_3[EOS]$" as input, then the model will autoregressively output the predicted segment "t_4t_5" until generates $[EOS]$ token.

C. TWO-STAGE TRAINING

As previously stated in Section III-B, the augmented data from the GEC corpus is beneficial in terms of generalizing the APR model. Therefore, as shown in Figure 3(a), we can train the APR model with a mixed training set containing gold data and augmented data:

$$\mathcal{D} = \mathcal{D}_{gold} \cup \mathcal{D}_{aug} \tag{6}$$

where $\mathcal{D}_{gold}$ and $\mathcal{D}_{aug}$ denote the gold and the augmented training data respectively. However, due to the fact that the GEC corpus is created from written language, the synthesized source transcript does not include numerous speech disfluencies and other faults that are common in spoken language. If we combine the augmented data and gold data during model training, as a consequence, the enormous augmented data has a tendency to overwhelm the gold data and add unneeded and even erroneous editing knowledge, which is detrimental to readability. In order to tackle this issue, we follow the lead of Zhang et al. [70] and train the model utilizing augmented data and gold data in two stages: pre-training and fine-tuning.

As shown in Figure 3(b), in the first stage we train the model with the augmented data $\mathcal{D}_{aug}$ using seq2seq learning:

$$\hat{\boldsymbol{\psi}} = \underset{\boldsymbol{\psi}}{\operatorname{argmax}} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}_{aug}} \log P(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\psi}) \tag{7}$$

where the $\hat{\boldsymbol{\psi}}$ denotes the parameters of the model learned in the pre-training stage. In the second stage we fine-tune the pre-trained model with the gold data $\mathcal{D}_{gold}$ using seq2seq learning:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}_{gold}} \log P(\boldsymbol{y} \mid \boldsymbol{x}, \hat{\boldsymbol{\psi}}; \boldsymbol{\theta}) \tag{8}$$

where the $\hat{\boldsymbol{\theta}}$ denotes the parameters of the model learned in the fine-tuning stage and is used during inference in Equation 5.

During the pre-training and fine-tuning stages of the process, the augmented data is not processed in the same way as the gold data. Instead, it just serves as prior knowledge that may be updated, if necessary, and even completely rewritten during the fine-tuning step. As a result, the model can learn more successfully from the gold data since it isn't distracted by the augmented data. Moreover, by segmenting the augmented and gold data into separate training stages, the model becomes more tolerant of noise in the augmented data, lowering the quality requirement for the augmented data and allowing the model to accept noisier augmented data, or even training data from other tasks, during the training stage. It is via this two-stage training process that the model not only learns critical information from the augmented data, but it also avoids being overloaded and adversely impacted by the augmented data.

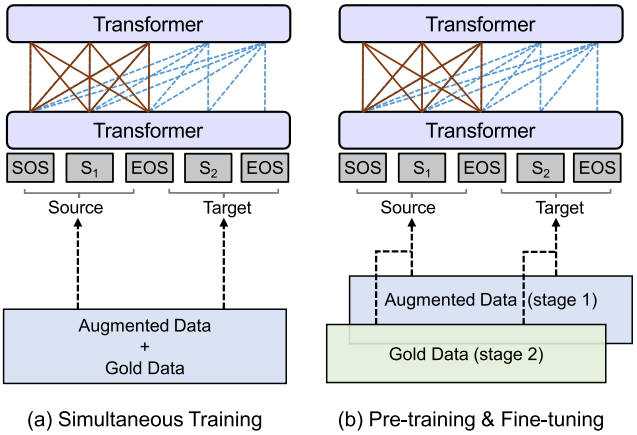

FIGURE 3. Comparison between simultaneous training and two-stage training.

V. EXPERIMENTAL SETTINGS

A. EVALUATION METRICS

In light of the fact that our goal is to increase the readability of automatic speech transcription, the word error rate (WER), a typical measure that is frequently used in speech recognition, is not appropriate for our application. Instead, as part of our research, we looked at the usefulness and consistency of many measures that are either directly or indirectly derived from related tasks such as speech recognition and machine translation, or that are modified from them.

- **Speech Recognition Metric** Our research first focused on extending the traditional WER in speech recognition to readability-aware WER (RA-WER) by omitting the text normalization step prior to computing the Levenshtein distance. We considered any mismatches owing to grammatical flaws, disfluency, as well as inappropriate capitalization, punctuation, and written numerical entity forms, to be errors. If there are other references, we chose the one that is the most similar to the candidate.
- **Machine Translation Metric** Alternatively, the APR task may be seen as a translation problem from a spoken transcript to a more easily understandable written text. As a result, we may make use of the Bilingual Evaluation

Understudy (BLEU) [71] score, which is frequently used in machine translation to assess the performance of the APR task in this scenario. In BLEU, the precision score is computed over variable-length of n-grams with length penalty [72] and optionally with smoothing [73].

B. BASELINE METHODS

In this work, we study the ASR post-processing for the readability task that aims to improve the readability of the ASR transcript. Therefore, we need baselines with the same or similar goal to the APR task. To the best of our knowledge, the most similar system with the goal of the APR task is inverse text normalization. Inverse Text Normalization (ITN) is the process of converting spoken text to its written form. ITN is commonly used to convert the output of an automatic speech recognition (ASR) system to increase the readability for users and automatic downstream processes [74]. Almost all commercial STT services incorporate the ITN to increase the readability for users and automatic downstream processes. We choose four STT services that provide the audio transcription as the baseline systems including an in-house Speech-to-Text service, Google Cloud Speech-to-Text service,[3] Microsoft Azure Speech-to-Text service,[4] and IBM Watson Speech-to-Text service.[5]

Note we take these STT services as a black box, which means that we input the audio file and get the readable text from the service output in one shot. On the contrary, our approach is applied in an explicit step-by-step manner, i.e. ASR and ASR post-processing, where the ASR system is the same hybrid model used for synthesizing the augmented data (Section III-B).

C. MODEL TRAINING

Based on the RoBERTa-large architecture (24-layer, 1024-hidden, 16-heads, 355M parameters), our model is developed using PyTorch on top of the Huggingface Transformers library.[6] We train our model using the Adam optimizer [75] ($\beta_1 = 0.9$, $\beta_2 = 0.999$) for 3 epochs and a batch size of 8 for each GPU. After warming up throughout the first tenth of all steps to a peak value of 2.5e-6, the learning rate linearly declines. For each model, 4 NVIDIA V100 GPUs with 32GB of memory and mixed-precision are used in the training process. In accordance with Vaswani et al. [49], we utilized label smoothing of 0.1 for regularization. Our vocabulary is the same as the 50K byte-level BPE vocabulary used by RoBERTa, which allows us to directly transfer its pre-trained weights. When it comes to both training phases, we employ the same training setting. In the fine-tuning stage, we choose checkpoints based on the validation set and set the beam size for beam search to 5 after searching in {1, 3, 5, 7}.

[3]https://cloud.google.com/speech-to-text

[4]https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/

[5]https://www.ibm.com/cloud/watson-speech-to-text

[6]https://github.com/huggingface/Transformers

TABLE 3. Performance of our approach and baseline systems on the test set of the APR data.

System	RA-WER↓	BLEU↑
1. Baseline systems		
In-house STT Service	38.77	53.39
Google Cloud STT Service	39.56	52.17
Microsoft Azure STT Service	38.18	54.20
IBM Watson STT service	40.54	51.59
2. ASR (hybrid model) + proposed APR model		
Our Approach	**20.26**	**74.29**
w/o pre-training	21.33	72.77

VI. RESULTS AND DISCUSSION

A. MAIN RESULTS

Table 3 shows the results of proposed APR models and baseline methods on the test set of the APR task.

In Table 3, we can see that all four speech-to-text services achieve comparable performance and lag much behind our approach. Our approach outperforms the in-house STT service baseline by a significant margin of 18.51 RA-WER points and 20.9 BLEU points (absolute value). This is reasonable because readability is not the eventual goal of the baseline systems, and they tend to transcribe the speech verbatim to ensure high recognition accuracy. On the contrary, the goal of the APR task is "translating" the ASR transcript into a highly readable text. Thus our approach trained on the APR dataset demonstrates the superiority of improving the readability of the ASR transcript.

We also conduct an ablation study on our approach by removing the pre-training stage in the proposed two-stage training strategy. Compared to the two-stage training (pre-training & fine-tuning), training the APR model without pre-training on the augmented data results in performance degradation, which proves the effectiveness of the proposed data augmentation method and the two-stage training strategy for the APR task.

B. ANALYSIS OF BASELINE SYSTEM

The in-house STT service is a pipeline composed of three components including a hybrid model, an n-best language model (LM), and an inverse text normalization (ITN). Using this pipeline can sequentially improve the accuracy of speech recognition and optimize the display format for readability.

The hybrid model used in in-house STT service is the same ASR model used for data augmentation (Section III-B). ASR models are often supplemented by separately trained language models that rescore the list of n-best hypotheses in order to improve the accuracy of speech recognition. Specifically, a stacked recurrent neural network with two unidirectional Long Short-Term Memory (LSTM) layers [76] is utilized as the language model in this pipeline. After decoding the ASR model output with beam search and rescoring the n-best list with the language model, the pipeline modifies the sentence display format for readability by invoking an ITN

TABLE 4. Results on the test set of the APR data between the output of each step of the in-house pipeline and the reference sentence.

Method	RA-WER↓	BLEU↑
Hybrid model (ASR transcript)	45.13	45.10
+ LM rescoring	44.71	45.68
+ ITN	38.77	53.39
+ Remove disfluencies	**33.52**	**56.76**

service. Specifically, a rule-based weighted finite-state transducer (WFST) is used in the pipeline due to low tolerance towards unrecoverable errors [77].

To better understand the contribution of each step of the pipeline to readability, we compute metric scores between the output of each step of the pipeline and the reference sentence in Table 4.

From the table, we can see that the LM rescoring only results in a 0.42 RA-WER decrease and a 0.58 BLEU promotion. In an extremely informal and conversational speech, recognition accuracy, particularly verbatim recognition, only accounts for a tiny proportion of the readability. Different from the APR task that aims to improve readability, LM rescoring focuses on improving the accuracy of speech recognition and overlooks factors related to readability, such as segmentation and disfluencies.

The ITN is one of the processes in the pipeline that is primarily concerned with increasing the readability of the ASR transcript. The ITN is made up of various modules, each of which performs a specific function. One module attempts to determine if a sentence is a statement (which requires the period "." at the end) or a question (which requires the question mark "?" at the end), and inserts proper punctuation where necessary. Moreover, it must figure out how to group together words to form sentences. Capitalization is handled by a separate module that is responsible for capitalizing names and the first word of sentences. By means of these processes, the ITN further improves the scores by 5.94 RA-WER and 7.71 BLEU, but not that far. It goes without saying that appropriate capitalization and punctuation are beneficial to readability, yet they are not sufficient.

Although the ITN improves the readability of the ASR transcript by mapping from spoken to written forms, it can not handle disfluencies in spoken language, which is especially common in spontaneous speech. To provide a fair comparison with our approach, we add an extra step to the pipeline for eliminating certain disfluencies. To be more specific, we remove the often used *filled pauses* (e.g., uh, um) and *discourse markers* (e.g., you know, I mean), and filter out the repeated words by keeping just one of them (e.g., I'm I'm→I'm, it's it's→it's). With these simple operations of removing some disfluencies, the performance gain is significant (RA-WER 5.25, BLEU 3.37), which implies that disfluencies are the major factor in the gap between the baselines and our approach.

TABLE 5. Results of using different models as the APR model. The ASR system used in this experiment is the hybrid model.

APR Model	RA-WER↓	BLEU↑
1. Use Seq2Seq models for the APR task		
LSTM	26.69	67.18
w/o pre-training	36.33	57.69
Transformer	24.04	69.26
w/o pre-training	38.69	55.51
2. Use pre-trained models for the APR task		
UniLM	23.44	71.76
w/o pre-training	24.63	70.02
Our Model	**20.26**	**74.29**
w/o pre-training	21.33	72.77

C. COMPARISON OF APR MODELS

In this subsection, we make a comparison of using different models to perform the APR task. Our proposed APR model uses the RoBERTa as the backbone and adapts it to a seq2seq model by using a specific self-attention mask. The UniLM also adapts the BERT to a seq2seq model with a particular self-attention mask. The main difference between our model and the UniLM is that we use an autoregressive approach to achieve whole-sentence prediction rather than only masked-position prediction during the training process, which utilizes training data more efficiently. We conduct an experiment to prove this in this subsection. Since both our model and the UniLM are pre-trained models, we also compare them with standard seq2seq models such as LSTM-based and Transformer-based.

We briefly introduce baselines used in the subsection as follows.

- LSTM-based seq2seq models are fundamental seq2seq models, which employ an LSTM [78] to map a sentence into a dense, fixed-length vector representation. LSTM, as opposed to RNN, is useful for handling long sequences, however it is unable to preserve the global information of the sequences.
- Transformer-based seq2seq models are cutting-edge seq2seq models that have been widely used to handle a variety of NLP tasks such as machine translation and abstractive summarization. Transformer [49] employs a self-attention mechanism that directly models the relationships between all words in an input sequence, independent of their order. Transformer, unlike LSTM, processes the full input sequence at once rather than iterating words one by one.
- UniLM is a pre-trained model using the BERT architecture and three types of language modeling tasks: unidirectional, bidirectional, and seq2seq prediction. The unified modeling approach is achieved by employing a shared Transformer network and utilizing specific self-attention masks to control what context the prediction conditions on. This allows the UniLM to be used for both discriminative and generative tasks.

As shown in Table 5, the pre-trained models (UniLM & our model) perform much better than the vanilla seq2seq models

(LSTM & Transformer). For all models, the performances degrade when excluding the pre-training stage of the proposed two-stage training strategy.

In the first group of Table, we can see that the Transformer outperforms the LSTM when using the proposed two-stage training. However, when training the Transformer directly on the APR data (w/o pre-training), it underperforms the LSTM without pre-training because of the problem of extreme overfitting on the relatively small training dataset. Pre-training the Transformer with the augmented data can overcome this problem, which brings a more significant improvement (BLEU score 69.26 vs. 55.51) than for pre-trained models like UniLM (BLEU score 71.76 vs. 70.02).

In the second group of Table, we can see that our model surpasses the UniLM, showing that the autoregressive approach exploits training data more efficiently.

D. COMPARISON OF ASR SYSTEMS

In this work, we mainly focus on studying the ASR post-processing for readability. Therefore, we use an off-the-shelf hybrid model for ASR to obtain the transcript for the input of the APR model throughout the whole work. Although we use the ASR system as is, it might be interesting to see how much gain our approach provides on top of different ASR systems. In this subsection we compare some end-to-end ASR systems on the APR task.

Different from hybrid models comprised of separate acoustic, pronunciation, and language modeling components that are trained independently [79], [80], end-to-end trained seq2seq ASR systems directly map the input acoustic speech signal to grapheme or word sequences [81], [82], [83], [84], [85]. In such seq2seq models, the acoustic, pronunciation, and language modeling components are trained jointly in a single system. Since these models directly predict graphemes or words, the process of decoding utterances is greatly simplified. Because seq2seq models are easier to train and require less human labor than a traditional approach, they are gaining new popularity in both research and industry settings.

In this subsection, we compare several predominant seq2seq models including connectionist temporal classification (CTC) [83], the recurrent neural network transducer (RNN-T) [82], attention-based encoder-decoder (or LAS: Listen, Attend and Spell [81]). The CTC criterion was proposed as a way of training end-to-end models without requiring a frame-level alignment of the target labels for a training utterance. The RNN-T augments the encoder network from the CTC model architecture with a separate recurrent prediction network over the output symbols. Because of its streaming nature, RNN-T has become a very promising end-to-end model in industry to replace the traditional hybrid models. Attention-based models like LAS consist of an encoder network, which maps the input acoustics into a higher-level representation, and an attention-based decoder that predicts the next output symbol conditioned on the full sequence of previous predictions.

TABLE 6. Results of using different ASR systems to obtain the transcript as the input for the APR task. The APR model used in this experiment is the proposed model in this work.

ASR System	RA-WER↓	BLEU↑
CTC	40.84	54.47
LAS	**19.89**	**75.44**
RNN-T	22.32	72.83
Hybrid model	20.26	74.29

As shown in Table 6, LAS, RNN-T, and Hybrid model get comparable results, while CTC is significantly lower than others. The main reason is that a CTC model is highly dependent on the use of an external language model to have acceptable accuracy. RNN-T and LAS models do not need an external language model due to the existence of a decoder component in the model. Among LAS, RNN-T, and Hybrid model, LAS outperforms Hybrid model while RNN-T underperforms Hybrid model. Previous works [86], [87] show a similar trend in the accuracy rate of speech recognition in terms of WER. Although the results in Table 6 are obtained through additional post-processing for readability, it can be seen that the performance of ASR systems is critical to downstream tasks such as the APR task. This inspires us that we can build an end-to-end speech recognition system for readability, which directly maps the input acoustic speech signal to highly readable text to avoid the error propagation between ASR and ASR post-processing steps. We leave it to future study.

E. HUMAN EVALUATION

Because readability is subjective, the BLEU score and the RA-WER may not be congruent with what people really perceive. Thus, we undertake a human evaluation on the Switchboard corpus [56] in addition to the automatic evaluation. Our A/B test, in particular, was conducted in order to compare our model with the baseline method. We created a test set for human evaluation by selecting 100 audio samples at random from a pool of source sentences ranging in length from 20 to 60 words. These audio samples are sent into the ASR system, which generates transcripts from them. After that, we construct the output text using both the baseline method (the in-house STT service) and our model, which are both described above. There are three annotators who are presented with the produced texts in a random sequence and asked to identify the one that is most readable. To avoid bias, the transcripts seen by the annotators were randomly shuffled, and the approach used to construct each transcript was kept a secret. Annotators were given the option of listening to the original audio if the produced text was difficult to understand. Each sample is assigned three labels by three different annotators. The ultimate decision is determined by a majority vote.

Annotators chose the output of our model 70 times out of 100 times (a win rate of 70%), indicating that our model is judged as more readable than the baseline method, according to the results of the human evaluation described above. This is confirmed by the two-sided binomial test on our results,

TABLE 7. Comparison of readable transcripts generated using the baseline method and the proposed model. The bold parts of sentences are corrections for recognition errors.

Input (ASR output)	yeah i don't believe they have to pay any uh like federal tax
Ground Truth	I don't believe , they have to pay any federal tax .
Baseline Method	Yeah, I don't believe they have to pay any. Uh, like federal tax, uh?
Our Approach	I don't believe, they have to pay any federal tax.
Input (ASR output)	yeah i i buy him every once **in awhile** and an i bought one and it was you know blah
Ground Truth	I buy them every once **in a while** . And I bought one . And it was blah .
Baseline Method	Yeah I I buy 'em every once **in awhile** and an I bought one and it was, you know blah.
Our Approach	I buy them every once **in a while**. And I bought one. And it was blah.
Input (ASR output)	they have as far as i'm concerned because i'm i'm not a big vegetable eater they have too many a yellow vegetables on the same day and
Ground Truth	They have too many yellow vegetables on the same day .
Baseline Method	They have, as far as I'm concerned, because I'm I'm not a big vegetable eater. They have too many a yellow vegetables on the same day and.
Our Approach	They have too many yellow vegetables on the same day.
Input (ASR output)	i just see a lot of a social and cultural differences that could **a post problems** with a puerto rico becoming a state
Ground Truth	I just see a lot of social and cultural differences that could **pose problems** with Puerto Rico becoming a state .
Baseline Method	I just see a lot of, uh, social and cultural differences that could **a post problems** with Puerto Rico becoming a state.
Our Approach	I just see a lot of social and cultural differences that could **cause problems** with Puerto Rico becoming a state.

which indicates that our model is statistically substantially more readable than the baseline method, with a p-value of less than 0.01.

F. CASE ANALYSIS

We compare the output samples of our approach with the baseline method (the in-house STT service) in order to undertake a qualitative study of readable transcripts generated by the proposed model. For reference, we also give the ground truth of the APR dataset constructed from the human-annotated MDE corpus (Section III-A). As seen in Table 7, both the baseline method and our approach enhance the readability of the ASR transcript by including punctuation and capitalizing the names and initial words of sentences, respectively. However, the baseline is maintained verbatim by including all of the words from the ASR transcript, which results in a disfluent sentence as well as improper segmentation and punctuation of the phrase. For example, in the first example, because of the impact of filler words (''uh'', ''like''), the baseline incorrectly splits the phrase and inserts a question mark instead of a period (''they have to pay any. Uh, like federal tax, uh?''). Our approach, in contrast to the baseline, eliminates any words that create disfluency and inserts accurate punctuation in the right places in the transcript to make the transcript more accessible and understandable. Using the third example, we can see how effective the proposed model is in eliminating disfluencies. For example, Our approach eliminates *asides and parentheticals* (''as far as i'm concerned because i'm i'm not a big vegetable eater'') to produce a coherent sentence that is consistent with the ground truth.

Our approach, in addition to correcting punctuation and capitalization problems, as well as reducing disfluencies, also corrects certain recognition errors, which the baseline method fails to accomplish. Table 7 has been highlighted with a bold type font to draw attention to the inaccuracies that have been fixed. This leads to an intriguing discovery: the final example has the term ''a post problems'' replaced with ''cause problems'', which is distinct from the term ''pose problems'' in the ground truth since the latter is not often used. While it is true that the original user input is ''pose'', we might argue that our model's output is better readable for the majority of human readers and machine applications if we do not take personalization into account.

VII. CONCLUSION

In this work, we study the problem of improving the readability of ASR output transcripts. We formulate the ASR post-processing for readability as a seq2seq text generation problem, which means we intend to obtain the human-readable text from the raw ASR output in one shot and without any extra information except the textual input.

We construct the dataset for this task from the MDE corpus. To overcome the problem of relatively small training data, we propose a novel data augmentation method that uses the TTS plus ASR to synthesize large-scale training data from GEC seed corpora. To make better use of the augmented data, we utilize a two-stage training strategy. We exploit an adapted RoBERTa pre-trained language model to perform the APR task, which can directly ''translate'' the ASR output to an error-free and readable transcript for human understanding and downstream tasks. We compare its performance with a pipeline-based baseline method deployed in the in-house speech-to-text service. Automatic and human evaluation demonstrate that our model outperforms the traditional pipeline-based baseline method and generates a more readable transcript.

ACKNOWLEDGMENT

An earlier version of this paper was presented in part at the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) in 2021 [DOI: 10.1109/ICASSP39728.2021.9414626].

REFERENCES

[1] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, ''The Microsoft 2017 conversational speech recognition system,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5934–5938.

[2] A. Ramani, A. Rao, V. Vidya, and V. B. Prasad, ''Automatic subtitle generation for videos,'' in *Proc. 6th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Mar. 2020, pp. 132–135.

[3] V. Aswin, M. Javed, P. Parihar, K. Aswanth, C. Druval, A. Dagar, and C. Aravinda, ''NLP-driven ensemble-based automatic subtitle generation and semantic video summarization technique,'' in *Advances in Artificial Intelligence and Data Engineering*. Cham, Switzerland: Springer, 2021, pp. 3–13.

[4] J. J. Koay, A. Roustai, X. Dai, and F. Liu, ''A sliding-window approach to automatic creation of meeting minutes,'' in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Student Res. Workshop*, 2021, pp. 68–75.

[5] M. Manuel, ''Automated generation of meeting minutes using deep learning techniques,'' *Int. J. Comput. Digit. Syst.*, vol. 2021, pp. 109–120, Jul. 2021.

[6] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, ''Learning phrase representations using RNN encoder–decoder for statistical machine translation,'' in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.

[7] J. Liao, Y. Shi, M. Gong, L. Shou, H. Qu, and M. Zeng, ''Improving zero-shot neural machine translation on language-specific encoders–decoders,'' in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.

[8] A. Paranjape and C. Manning, ''Human-like informative conversations: Better acknowledgements using conditional mutual information,'' in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2021, pp. 768–781.

[9] S. Bao, H. He, F. Wang, H. Wu, and H. Wang, ''PLATO: Pre-trained dialogue generation model with discrete latent variable,'' in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 85–96.

[10] C. Shan, J. Zhang, Y. Wang, and L. Xie, ''Attention-based end-to-end speech recognition on voice search,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4764–4768.

[11] Y. Y. Wang, D. Yu, Y. C. Ju, and A. Acero, ''An introduction to voice search,'' *IEEE Signal Process. Mag.*, vol. 25, no. 3, pp. 28–38, May 2008.

[12] A. Chuklin, A. Severyn, J. R. Trippas, E. Alfonseca, H. Silen, and D. Spina, ''Using audio transformations to improve comprehension in voice question answering,'' in *Proc. Int. Conf. Cross-Lang. Eval. Forum Eur. Lang.* New York, NY, USA: Springer, 2019, pp. 164–170.

[13] P. Rosso, L.-F. Hurtado, E. Segarra, and E. Sanchis, ''On the voice-activated question answering,'' *IEEE Trans. Syst., Man, Cybern., C, Appl. Rev.*, vol. 42, no. 1, pp. 75–85, Jan. 2012.

[14] S. Strassel, C. Walker, and H. Lee, ''RT-03 MDE training data speech,'' in *Linguistic Data Consortium*. Philadelphia, PA, USA: Springer, 2004.

[15] S. Strassel. (May 14, 2003). *Simple Metadata Annotation Specification Version 5.0*. [Online]. Available: https://catalog.ldc.upenn.edu/docs/LDC2004S08/SimpleMDE_V5.0.pdf

[16] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, ''RoBERTa: A robustly optimized BERT pretraining approach,'' 2019, *arXiv:1907.11692*.

[17] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, ''Unified language model pre-training for natural language understanding and generation,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 13063–13075.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, ''BERT: Pre-training of deep bidirectional transformers for language understanding,'' in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186.

[19] J. Liao, Y. Shi, M. Gong, L. Shou, S. Eskimez, L. Lu, H. Qu, and M. Zeng, ''Generating human readable transcript for automatic speech recognition with pre-trained language model,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 7578–7582.

[20] R. Errattahi, A. El Hannani, and H. Ouahmane, ''Automatic speech recognition errors detection and correction: A review,'' *Proc. Comput. Sci.*, vol. 128, pp. 32–37, May 2018.

[21] J. Guo, T. N. Sainath, and R. J. Weiss, ''A spelling correction model for End-to-end speech recognition,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brighton, U.K., May 2019, pp. 5651–5655.

[22] O. Hrinchuk, M. Popova, and B. Ginsburg, ''Correction of automatic speech recognition with transformer sequence-to-sequence model,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 7074–7078.

[23] M. Paulik, S. Rao, I. Lane, S. Vogel, and T. Schultz, ''Sentence segmentation and punctuation recovery for spoken language translation,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2008, pp. 5105–5108.

[24] E. Cho, J. Niehues, and A. Waibel, ''Segmentation and punctuation prediction in speech language translation using a monolingual translation system,'' in *Proc. Int. Workshop Spoken Lang. Transl. (IWSLT)*, 2012, pp. 252–259.

[25] E. Cho, J. Niehues, K. Kilgour, and A. Waibel, ''Punctuation insertion for real-time spoken language translation,'' in *Proc. 12th Int. Workshop Spoken Lang. Transl., Papers*, 2015, pp. 1–8.

[26] F. Batista, D. Caseiro, N. Mamede, and I. Trancoso, ''Recovering capitalization and punctuation marks for automatic speech recognition: Case study for Portuguese broadcast news,'' *Speech Commun.*, vol. 50, no. 10, pp. 847–862, Oct. 2008.

[27] A. Gravano, M. Jansche, and M. Bacchiani, ''Restoring punctuation and capitalization in transcribed speech,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2009, pp. 4741–4744.

[28] S. Škodová, M. Kuchařová, and L. Šeps, ''Discretion of speech units for the text post-processing phase of automatic transcription (in the Czech language),'' in *Proc. Int. Conf. Text, Speech Dialogue*. Cham, Switzerland: Springer, 2012, pp. 446–455.

[29] S. Rothe, J. Mallinson, E. Malmi, S. Krause, and A. Severyn, ''A simple recipe for multilingual grammatical error correction,'' in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 702–707.

[30] C. Napoles, M. Nădejde, and J. Tetreault, ''Enabling robust grammatical error correction in new domains: Data sets, metrics, and analyses,'' *Trans. Assoc. Comput. Linguistics*, vol. 7, pp. 551–566, Nov. 2019.

[31] Y. J. Choe, J. Ham, K. Park, and Y. Yoon, ''A neural grammatical error correction system built on better pre-training and sequential transfer learning,'' in *Proc. 14th Workshop Innov. NLP Building Educ. Appl.* Florence, Italy: Association for Computational Linguistics, 2019, pp. 252–263.

[32] M. Mita, T. Mizumoto, M. Kaneko, R. Nagata, and K. Inui, ''Cross-corpora evaluation and analysis of grammatical error correction models—Is single-corpus evaluation enough?'' in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 1309–1314.

[33] F. Wang, W. Chen, Z. Yang, Q. Dong, S. Xu, and B. Xu, ''Semi-supervised disfluency detection,'' in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 3529–3538.

[34] P. J. Lou, Y. Wang, and M. Johnson, "Neural constituency parsing of speech transcripts," in *Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2019, pp. 2756–2765.
[35] M. Shugrina, "Formatting time-aligned ASR transcripts for readability," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol.* Los Angeles, CA, USA: Association for Computational Linguistics, 2010, pp. 198–206.
[36] Y. Liu, E. Shriberg, A. Stolcke, B. Peskin, J. Ang, D. Hillard, M. Ostendorf, M. Tomalin, P. Woodland, and M. Harper, "Structural metadata research in the EARS program," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2005, pp. 102–150.
[37] C. Wayne, "Effective, affordable, reusable speech-to-text (ears)," DARPA, Arlington, VA, USA, Tech. Rep., 2003. [Online]. Available: http://www.darpa.mil/iao/EARS.htm
[38] Y. Liu, E. Shriberg, A. Stolcke, B. Peskin, and M. Harper, "The ICSI/SRI/UW RT-04 structural metadata extraction system," in *Proc. EARS RT-04 Workshop*, 2004, pp. 1–5.
[39] M. Tomalin and P. Woodland, "The RT04 evaluation structural metadata systems at cued," in *Proc. Fall Rich Transcription Workshop (RT-04)*, 2004, pp. 110–145.
[40] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper, "Comparing and combining generative and posterior probability models: Some advances in sentence boundary detection in speech," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2004, pp. 64–71.
[41] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, vol. 28, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Montreal, QC, Canada: Springer, Dec. 2015, pp. 3079–3087.
[42] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds. Long Beach, CA, USA: Springer, Dec. 2017, pp. 6294–6305.
[43] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1. Melbourne, VIC, Australia: Association for Computational Linguistics, 2018, pp. 328–339.
[44] T. Brown, "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1877–1901.
[45] O. Lieber, O. Sharir, B. Lenz, and Y. Shoham, "Jurassic-1: Technical details and evaluation," AI21 Labs, Tel Aviv, Israel, White Paper, 2021. [Online]. Available: https://www.ai21.com/blog/announcing-ai21-studio-and-jurassic-1
[46] S. Smith, M. Patwary, B. Norick, P. LeGresley, S. Rajbhandari, J. Casper, Z. Liu, S. Prabhumoye, G. Zerveas, V. Korthikanti, E. Zhang, R. Child, R. Y. Aminabadi, J. Bernauer, X. Song, M. Shoeybi, Y. He, M. Houston, S. Tiwary, and B. Catanzaro, "Using DeepSpeed and megatron to train megatron-turing NLG 530B, a large-scale generative language model," 2022, *arXiv:2201.11990*.
[47] J. W. Rae, "Scaling language models: Methods, analysis & insights from training gopher," 2021, *arXiv:2112.11446*.
[48] A. Chowdhery, "Palm: Scaling language modeling with pathways," 2022, *arXiv:2204.02311*.
[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[50] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. (2018). *Improving Language Understanding by Generative Pre-Training*. [Online]. Available: https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/languageUnderstand.paper.pdf
[51] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," OpenAI, San Francisco, CA, USA, 2019, vol. 1, no. 8. [Online]. Available: https://openai.com/blog/better-language-models
[52] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "XlNet: Generalized autoregressive pretraining for language understanding," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds. Vancouver, BC, Canada: Springer, Dec. 2019, pp. 5754–5764.
[53] K. Song, X. Tan, T. Qin, J. Lu, and T. Liu, "MASS: Masked sequence to sequence pre-training for language generation," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, vol. 97, K. Chaudhuri and R. Salakhutdinov, Eds. Long Beach, CA, USA: Springer, 2019, pp. 5926–5936.
[54] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880.
[55] 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," 2019, *arXiv:1910.10683*.
[56] J. J. Godfrey and E. Holliman, "Switchboard-1 release 2," in *Linguistic Data Consortium*, vol. 926. Philadelphia, PA, USA: Springer, 1997, p. 927.
[57] T. Ge, F. Wei, and M. Zhou, "Reaching human-level performance in automatic grammatical error correction: An empirical study," 2018, *arXiv:1807.01270*.
[58] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4779–4783.
[59] Y. Deng, L. He, and F. Soong, "Modeling multi-speaker latent space to improve neural TTS: Quick enrolling new speaker and enhancing premium voice," 2018, *arXiv:1812.05253*.
[60] J. Li, R. Zhao, E. Sun, J. H. M. Wong, A. Das, Z. Meng, and Y. Gong, "High-accuracy and low-latency speech recognition with two-head contextual layer trajectory LSTM model," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 7699–7703.
[61] Y. Miao, J. Li, Y. Wang, S.-X. Zhang, and Y. Gong, "Simplifying long short-term memory acoustic models for fast training and decoding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 2284–2288.
[62] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden Markov models for speech recognition," *Comput. Speech Lang.*, vol. 16, no. 1, pp. 25–47, Jan. 2002.
[63] J. H. M. Wong and M. J. F. Gales, "Sequence student-teacher training of deep neural networks," in *Proc. Interspeech*, Sep. 2016, pp. 134–155.
[64] C. Bryant, M. Felice, Ø. E. Andersen, and T. Briscoe, "The BEA-2019 shared task on grammatical error correction," in *Proc. 14th Workshop Innov. NLP Building Educ. Appl.*, 2019, pp. 1–4.
[65] H. Yannakoudakis, T. Briscoe, and B. Medlock, "A new dataset and method for automatically grading Esol texts," in *Proc. ACL*, 2011, pp. 180–189.
[66] T. Mizumoto, M. Komachi, M. Nagata, and Y. Matsumoto, "Mining revision log of language learning SNS for automated Japanese error correction of second language learners," in *Proc. IJCNLP*, Nov. 2011, pp. 147–155.
[67] T. Tajiri, M. Komachi, and Y. Matsumoto, "Tense and aspect error correction for ESL learners using global context," in *Proc. ACL*, 2012, pp. 1–5.
[68] S. Granger, "The computer learner corpus: A versatile new source of data for sla research: Sylviane Granger," in *Learner English on Computer*. London, U.K.: Routledge, 2014, pp. 25–40.
[69] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
[70] Y. Zhang, T. Ge, F. Wei, M. Zhou, and X. Sun, "Sequence-to-sequence pre-training with data augmentation for sentence rewriting," 2019, *arXiv:1909.06002*.
[71] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.
[72] K. Papineni, "Machine translation evaluation: N-grams to the rescue," in *Proc. LREC*, 2002, pp. 190–198.
[73] C.-Y. Lin and F. J. Och, "Orange: A method for evaluating automatic evaluation metrics for machine translation," in *Proc. 20th Int. Conf. Comput. Linguistics (COLING)*, 2004, pp. 501–507.
[74] M. Sunkara, C. Shivade, S. Bodapati, and K. Kirchhoff, "Neural inverse text normalization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 7573–7577.
[75] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, vol. 2015, Y. Bengio and Y. LeCun, Eds. San Diego, CA, USA: Springer, May 2015, pp. 1–3.
[76] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Proc. 13th Annu. Conf. Int. Speech Commun. Assoc.*, 2012, pp. 1892–1898.

[77] P. Ebden and R. Sproat, "The kestrel TTS text normalization system," *Natural Lang. Eng.*, vol. 21, no. 3, pp. 333–353, May 2015.
[78] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[79] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. INTERSPEECH*, H. Li, H. M. Meng, B. Ma, E. Chng, and L. Xie, Eds. Singapore: ISCA, Sep. 2014, pp. 338–342.
[80] T. N. Sainath, O. Vinyals, A. W. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, South Brisbane, QL, Australia, Apr. 2015, pp. 4580–4584.
[81] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 4960–4964.
[82] A. Graves, "Sequence transduction with recurrent neural networks," 2012, *arXiv:1211.3711*.
[83] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, vol. 148, W. W. Cohen and A. W. Moore, Eds. Pittsburgh, PA, USA: ACM, Jun. 2006, pp. 369–376.
[84] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, vol. 28, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Montreal, QC, Canada: ACM, Dec. 2015, pp. 577–585.
[85] L. Lu, X. Zhang, and S. Renals, "On training the recurrent neural network encoder–decoder for large vocabulary end-to-end speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process., (ICASSP)*, Shanghai, China, Mar. 2016, pp. 5060–5064.
[86] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *Proc. Interspeech*, F. Lacerda, Ed. Stockholm, Sweden: ISCA, Aug. 2017, pp. 939–943.
[87] T. N. Sainath, R. Pang, D. Rybach, Y. He, R. Prabhavalkar, W. Li, M. Visontai, Q. Liang, T. Strohman, Y. Wu, I. McGraw, and C. Chiu, "Two-pass end-to-end speech recognition," in *Proc. Interspeech*, vol. 2019, G. Kubin and Z. Kacic, Eds. Graz, Austria: ISCA, Sep. 2019, pp. 2773–2777.

JUNWEI LIAO is currently pursuing the Ph.D. degree in computer science and technology with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China.

His current research interests include neural networks, deep learning, and natural language processing.

YU SHI received the bachelor's degree in automation from Tsinghua University and the master's and Ph.D. degrees in pattern recognition and intelligent systems.

She is with Microsoft Cognitive Service Research Group. Her research interests include language model pretraining, machine translation, speech recognition, spoken language understanding, question answering, multilingual, and multimodality.

YONG XU received the graduate degree in computer science and engineering from the University of Electronic Science and Technology of China, in 2000.

Since 2000, he has been engaged in network security technology, product research and development, program design, network security services. His research interests include the application of artificial intelligence technology in the field of network security to improve the defense ability of unknown security threats and the effectiveness of simulated network attacks.