MULTI TRANSCRIPTION-STYLE SPEECH TRANSCRIPTION USING ATTENTION-BASED ENCODER-DECODER MODEL

Yan Huang, Piyush Behre, Guoli Ye, Shawn Chang, and Yifan Gong

Microsoft Corporation

ABSTRACT

Human professional transcription services provide a variety of transcription styles to customize different needs. To accommodate different users and facilitate seamless integration with downstream applications, we propose a framework to generate multi-style transcription in an attention-based encoder-decoder model (AED) using three different architectures: (A) style-dependent layers; (B) mixed-style output; (C) style-dependent prompt. In this framework, both the verbatim lexical transcription and the readable transcription of various styles can be generated simultaneously or separately, through a single decoding pass or multiple decoding passes on-demand. We conduct experiments in a large-scale AED-based speech transcription system trained with 50k hours speech. The proposed framework can achieve nearly on-par performance compared to the single-style AED with significant savings in model footprint and decoding cost. Moreover, it provides an efficient data sharing mechanism across different styles through knowledge transfer.

***Index Terms*—** attention-based encoder-decoder model (AED), verbatim lexical transcription, readable transcription

1. INTRODUCTION

Human professional speech transcription services provide a variety of transcription styles [1], ranging from phonetic transcription to intelligent edited transcription, to address varying customer needs. Traditional automated speech transcription systems usually produce unformatted **verbatim lexical transcription**. It is subsequently converted to the display format with proper punctuation, capitalization, and inverse text normalization (ITN) through a display formatting process (DPP) [2–7]. For conversation speech, disfluency and grammatical errors are removed to further improve readability [8–10]. We refer to this type of transcription as **readable transcription**.

Inferring the readable transcription merely from the unformatted text is not always sufficient. For instance, speech prosodic features were found to be helpful in improving sentence boundary detection [8, 11]. Instead of pursuing this two-stage approach, end-to-end models (E2E) [12–19], jointly modeling the acoustic and language dependencies, provide a promising way to produce the readable transcription end-to-end. One of the first attempt to generate the readable transcription directly from speech was reported on the Earnings call transcription task [20]. Whisper extended the success to a domain-independent large-scale multi-lingual setup [21]. Examples of some recent work include [22, 23].

Despite the recent success of Whisper, in a practical speech recognition service, different applications may require different types of transcription. For example, an automated closed-captioning service usually adopts the verbatim transcription rendered in the display format; nevertheless, a meeting summarization system would prefer the readable transcription with disfluency removed. To customize the need from different users and facilitate seamless integration with downstream applications, we propose a framework for generating the multi-style speech transcription in an attention-based encoder-decoder model (AED). In this framework, both the verbatim lexical transcription and the readable transcription of various styles can be generated simultaneously or separately, through a single-decoding pass or multiple decoding passes on-demand.

Specifically, we propose three multi-style AED architectures with style-dependent layers, mixed-style output, and style-dependent prompt. **The AED with style-dependent layers** uses a stack of style-dependent layers for each individual style, while keeping the rest network components shared [24, 25]. At decoding time, we only need to combine the shared network with the style-dependent branch to generate the transcription of a particular style. **The AED with mixed-style output** was inspired by the joint decoding and translation model [26–28]. In this approach, we use a single AED to generate the mixed-style output decorated with style tags. During training, the mixed-style transcription can be organized by concatenating the utterance-level or token-level multi-style transcription with style tags. At decoding time, mixed-style transcription can be generated through a single decoding pass. The style-dependent transcription can be subsequently extracted from it using a simple style decoder. In the third architecture, we use a single **AED with the style-dependent prompt** inserted at the decoder to generate different style transcription. In addition, one-hot style embedding can be inserted at various encoder and decoder layers. Here multiple decoding passes are needed to generate multi-style transcription.

We conduct experiments on a large-scale speech transcription system trained with 50k hours speech with both lexical and readable transcription. We use the token error rate (TER) and the segmentation F-measure to evaluate the performance of the readable transcription, and the word error rate (WER) for the verbatim lexical transcription. We found that the end-to-end readable AED outperforms the two-stage based approach with a lexical AED followed by the display format processing. This confirms that leveraging speech audio helps in generating transcription with improved readability. Second, the proposed multi-style AED can achieve nearly on-par performance compared to multiple single-style AEDs. Nevertheless, the total number of model parameters and the run-time cost are significantly smaller. Lastly, the AEDs with style-dependent layers or style-dependent prompt can facilitate knowledge transfer across styles. This is especially beneficial when training data is scarce.

To the best of our knowledge, this is the first proposed work in generating both verbatim lexical and readable multi-style transcription in an end-to-end system. A highly relevant work can be found in [29], which primarily focuses on generating rich transcription with laugh, cough, and other acoustic events.

The rest of this paper is organized as follows: Section 2 introduces the multi-style AED framework; Section 3 presents the experiments and results; Section 4 concludes the paper.

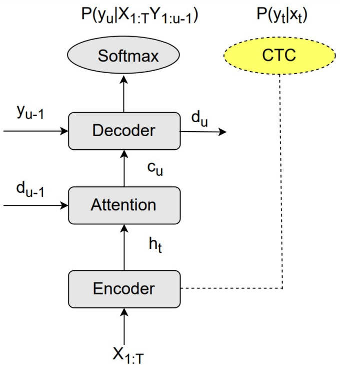


Fig. 1. Diagram of attention-based encoder-decoder model.

2. METHODOLOGY

In this section, we first introduce the foundation of AED, then present the architecture of the multi transcription-style AEDs.

2.1. Foundations of AED model

The attention-based encoder-decoder model, as depicted in Fig. 1, consists of an encoder, decoder, and an attention network [30, 31]. The encoder network converts the input feature sequence ($X_{1:T}$) to the hidden feature sequence (h_t). The attention module computes the attention weights between the previous decoder output ($d_{1:u-1}$) and the encoder output (h_t) using an attention function. These weights are then used to compute a context vector (c_u) as weighted sum of the encoder feature sequence (h_u). The decoder network takes the context vector (c_u) and the previous output label (y_{u-1}) to compute $p(y_u|x, y_{1:u-1})$. The final output is obtained by minimizing $-\log p(y_u|x, y_{1:u-1})$. To mitigate the alignment issue, an AED is often optimized together with CTC as a multi-task learning task [32].

Next we will describe the proposed multi transcription-style AEDs depicted in Fig. 2. To illustrate the proposed approach, we focus on modeling two styles (the verbatim lexical and the readable transcription) throughout this paper, though the proposed approach is general and can be extended to more styles.

2.2. Multi-Style AED with Style-Dependent Layers

The multi-style AED with style-dependent layers is an AED with style-dependent network branches. It utilises a stack of style-dependent top encoder layers (*encoder1*) and decoder to map the low-level speech features to different styles of transcription as in Fig. 2 (a). The bottom encoder layers (*encoder0*), located closer to the raw speech input layer, performing the low-level feature mapping, are believed to be mostly agnostic to styles. Such an architecture can be jointly trained with the combined objective from each style. Alternatively, one can train a style-agnostic model first, then fine-tune the style-dependent layers separately.

The model footprint and the decoding run-time cost are determined by where to branch out the style-dependent layers. The more the different styles can share, the smaller the additional run-time cost is needed to generate multi-style transcription. This architecture is flexible in consuming training data with different styles of transcription. Data in a certain style can be used to improve its own style branch with the potential to be transferred to the others through the shared style-agnostic bottom layers. We will present detailed experimental results in Section 3.3.

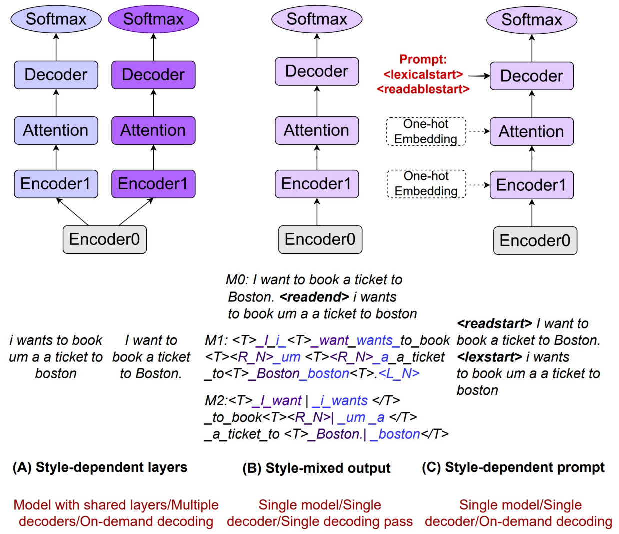

Softmax
Softmax
Decoder
Decoder
Attention
Attention
Encoder1
Encoder1
Encoder0
i wants to book um a a ticket to boston
I want to book a ticket to Boston.
(A) Style-dependent layers
Model with shared layers/Multiple decoders/On-demand decoding
Softmax
Decoder
Attention
Encoder1
Encoder0
M0: I want to book a ticket to Boston. <readend> i wants to book um a a ticket to boston
M1: <T>_I_i_<T>_want_wants_to_book <T><R_N>_um <T><R_N>_a_a_ticket _to<T>_Boston_boston<T>.<L_N>
M2:<T>_I_want | _i_wants </T> _to_book<T><R_N>| _um _a </T> _a_ticket_to <T>_Boston.| _boston</T>
(B) Style-mixed output
Single model/Single decoder/Single decoding pass
Softmax
Prompt: <lexicalstart> <readablestart>
Decoder
One-hot Embedding
Attention
One-hot Embedding
Encoder1
Encoder0
<readstart> I want to book a ticket to Boston. <lexstart> i wants to book um a a ticket to boston
(C) Style-dependent prompt
Single model/Single decoder/On-demand decoding

Fig. 2. Architecture of the proposed multi transcription-style AEDs with: (A) style-dependent layers; (B) mixed-style output; (C) style-dependent prompt. *Encoder0*: style-agnostic bottom encoder layers, *Encoder1*: style-dependent top encoder layers.

2.3. Multi-Style AED with Mixed-Style Output

In the second architecture, we use a standard AED to generate the mixed-style transcription simultaneously as in Fig. 2 (b). The mixed-style transcription can be composed in different ways with the introduction of a new descriptive language for multi-style transcription.

The concatenated mixed-style transcription $M0$ simply concatenates the different style utterance-level transcription with style tags:

$$U_{s_1} \langle T_{s_1} \rangle \; U_{s_2} \langle T_{s_2} \rangle, \tag{1}$$

where U is the utterance-level transcription, $\langle T \rangle$ is the style-end tag, s_i is the style index. An example of $M0$ is illustrated in Fig. 2 (b). It roughly doubles the transcription length (in case of two styles), which is inefficient especially when modeling more than two styles.

The interlaced mixed-style transcription is designed to reduce the length of the mixed-style transcription. Specifically, we first align the token-level multi-style transcription (u), then linearize the aligned transcription at each alignment location (L_j) according to a pre-defined style order (s_i):

$$u^{L_1}_{s_1} u^{L_1}_{s_2} \; u^{L_2}_{s_1} u^{L_2}_{s_2} \cdots \mathbf{u^{L_j}_{s_1} u^{L_j}_{s_2}}. \tag{2}$$

To compose $M1$, at each alignment location L_j, if different styles share the same transcription, simply include one copy of the shared transcription; otherwise, add different style transcription in the pre-defined order with a leading style-alternative tag $\langle T \rangle$:

$$u^{L_{j-1}} \; \langle \mathbf{T} \rangle \mathbf{u^{L_j}_{s_1} u^{L_j}_{s_2}} \; u^{L_{j+1}}. \tag{3}$$

$M2$ is an extension of $M1$. In $M2$, the contiguous $\langle T \rangle$-tagged segments in Eq. 3 is further merged into a style alternative section delimited by $\langle T \rangle$ and $\langle /T \rangle$. Inside this section, transcription of different styles are separated by a style separator tag (|), formally

$$u^{L_{j-1}} \; \langle \mathbf{T} \rangle \mathbf{u^{L_{j:j+m}}_{s_1}} | \mathbf{u^{L_{j:j+m}}_{s_2}} \langle /\mathbf{T} \rangle \; u^{L_{j+m+1}}. \tag{4}$$

An example of $M1$ and $M2$ can be found in Fig. 2 (b).

To extract the style-dependent transcription, we only need to write a simple style parser to decode the mixed-style decoding result

Table 1. Comparison of the proposed multi-style AEDs with: (A) style-dependent layers; (B) mixed-style output; (C) style-dependent prompt.

Type	Model	#Decoders	#Decoding Passes	Tag Inference	Data Consumption	#Styles (> 2)
(A) Style-dependent layers	Branched AED	Multi	Multi	No	Flexible	Yes
(B) Style-mixed output	Single AED	Single	Single	Yes	Co-existence	Limited
(C) Style-dependent prompt	Single AED	Single	Multi	No	Flexible	Yes

following the same protocol used to compose the training transcription. As the style tags are to be predicted at the inference time, we need to keep the tags as simple and error-tolerant as possible. To further improve the robustness of the style parser, we introduce a simple mechanism to ignore the erroneous tags, thus the style parser can always resume from the next legitimate style session. Empirically we found the tag prediction is highly accurate.

One particular aspect of this approach is that it requires all interested styles of transcription be available; otherwise the mixed-style transcription could not be properly composed during training.

It is to be noted that we focus on the non-streaming AED in this paper. Nevertheless, the interlaced mixed-style architecture can be extended to a streaming model such as the conformer transducer. Generating multi-style transcription simultaneously in a streaming model is appealing for applications where multi-style transcription is always desired.

2.4. Multi-Style AED with Style-Dependent Prompt

In the third architecture, we use a single AED with style-dependent prompts inserted at the beginning of the decoder to generate multi-style transcription as in Fig. 2 (c). During training, we create multiple instances of a training utterance by inserting a style prompt at the beginning of different style transcription:

$$\begin{array}{c}\langle T_{s_1}\rangle U_{s_1}\\ \langle T_{s_2}\rangle U_{s_2}\end{array} \tag{5}$$

An example of the style-dependent prompt is provided in Fig. 2 (c).

At decoding time, to generate the transcription of a specific style s_i, we simply add a style prompt $\langle T_{s_i}\rangle$ at the beginning of the decoder, then the decoder continues to decode the transcription in style s_i. We note that tags in this architecture are not to be predicted, instead they are provided as prompts to the system. In addition to the prompt, we can add one-hot embedding at different encoder and decoder layers to explicitly model different styles.

To generate transcription of multiple styles, multiple decoding passes are needed. Nevertheless, the total run-time cost is reduced as the encoder is shared across different decoding passes. Should one-hot embedding of different styles be inserted at the encoder layers, re-computing some encoder layers for different styles would be needed at a cost of computation efficiency.

Regarding the training data usage, the data consumption is fairly flexible as it does not require co-existence of all interested styles of transcription. The model is maximally shared across different styles. Another distinct property of this architecture is that it can be conveniently extended to multiple styles (e.g. > 2), as compared to the mixed-style output AED, where the transcription length increase poses a limitation on its extension to many styles.

2.5. Three Approaches in Comparison

We compare the proposed three approaches in Table 1 from the model structure, number of decoders, number of decoding passes needed to generate multi-style transcription, whether it involves tag inference, how flexible it is in consuming multi-style transcription data, and whether it can be extended to more styles conveniently.

The AEDs with style-dependent layers and with style-dependent prompt share several common aspects. For example, they both require multiple decoding passes to generate different style transcription. No tag inference is needed during decoding. They are both flexible in data consumption and can be conveniently extended to more styles. A major difference between them is that the latter uses a single decoder for all styles. The AED with the mixed-style output differs from the above two in many aspects. In particular, this model can generate multi-style transcription within a single decoding pass and therefore has the most benefit in run-time cost. The limitation of this approach is that it is less flexible in data consumption and less convenient when extending to many more styles.

In practice, the model complexity, model footprint, decoding cost, flexibility in data usage, capability of extending to more styles, and most of all the salient customer need are important considerations we take when choosing a specific architecture.

3. EXPERIMENTS AND RESULTS

In this section, we present experiments and results.

3.1. Training Data and Metrics

The training data consists of 50k hours anonymized speech with personal information removed. The verbatim lexical model was trained on the verbatim lexical transcription, which is provided by professional speech transcribers. To train the readable style model, we apply the internally developed DPP processing [6, 7] to convert the verbatim lexical transcription to its readable form, with proper capitalization, punctuation, and ITN. The editing disfluency, such as hesitation and filler words, are also removed.

For evaluation, we use two internally collected anonymized test sets with both verbatim lexical and readable transcription provided by professional transcribers. Test set A consists of 5 hours dictation monologue speech; Test set B consists of 4 hours conversation speech from real meetings.

For metrics, we use the token error rate (TER) to measure the performance of the readable transcription and the traditional word error rate (WER) for the verbatim lexical transcription. TER is the token-level editing distance between the hypothesis and the **readable reference**, without applying any forms of text normalization. In addition, to calibrate segmentation quality in readable transcription, we map a set of sentence ending punctuations (e.g. ;.?!) to a common segmentation mark, then compute the F-measure of the mapped segmentation mark. Properly segmenting a word sequence into sentences or sub-sentences is one of the most important factors affecting readability.

3.2. Experimental Setup and Baseline Models

The various AED models studied in this paper share the following similar setup: the audio encoder consists of two convolutional layers that sub-sample the time frame by a factor of 4, followed by 18 conformer layers. Each conformer layer has a multi-head attention with 8 heads, and a depth-wise convolution with kernel size of 3. The multi-head attention and the depth-wise convolution are sandwiched between two 1024-dim feed-forward layers. The decoder consists of 6 conformer layers and the feed-forward layer dimension is 2048. The embedding dimension is 512. The AED is trained to optimize the combined cross-entropy loss and the CTC loss (weighted by 0.2).

We train a pair of a verbatim lexical AED ($s.L$) and a readable AED ($s.R$) using the 50k hour training data as our baseline models. To simulate the traditional two-stage based approach, we apply the DPP post-processing to convert the lexical hypothesis to its readable form and thus obtain the two-stage based readable hypothesis ($s.L + DPP$).

The end-to-end readable model ($s.R$) outperforms the two-stage based approach ($s.L + DPP$) on both the dictation (A) and the conversation (B) tasks, as shown in Table 2. This suggests that it is beneficial to learn the readable transcription directly from speech as compared to only using the textual information in the two-stage based approach. In particular, the improvement in segmentation F-measure confirms that leveraging the speech-level information helps in improving segmentation. This can be seen in some decoding examples in Section 3.6.

Table 2. Performance of the single-style AEDs: $s.L$ is the verbatim lexical AED, $s.R$ is the readable AED, $s.L + DPP$ is the two-stage based approach with the DPP post-processing.

	A			B		
Model	TER	F1	WER	TER	F1	WER
$s.L$	NA	NA	6.2	NA	NA	9.0
$s.L + DPP$	17.6	0.65	NA	30.0	0.51	NA
$s.R$	16.7	0.71	NA	28.4	0.60	NA

We also made an interesting observation that, although the readable training transcription was generated from the lexical transcription using the DPP process, the resulting readable model ($s.R$) outperforms the two-stage based approach ($s.L + DPP$), applying the same DPP post-processing to convert the lexical hypothesis to its readable form during test. We believe this is primarily due to the benefit of end-to-end modeling of the acoustic and language dependency in deriving the readable transcription.

3.3. Result of Multi-Style AED with Style-Dependent Layers

In this section, we discuss the multi-style AED experiments. Table 3 summarizes the multi-style AED results with style-dependent layer ($m.L$), mixed-style output ($m.M$), and style-dependent prompt ($m.P$).

For the mixed-style AED with style-dependent layers, we experimented with branching out the style-dependent network component at different depth of the encoder layers. $m.L_{E_i}$ refers to a mixed-style AED with style-dependent layer starting from the encoder layer E_i. For example, in $m.L_{E6}$, the bottom encoder layers $E1 \sim E6$ are style-agnostic layers and the top encoder layer $E7 \sim E18$ and the decoder are style-dependent layers; in $m.L_{E18}$, all the encoder layers are style-agnostic layers; in $m.L_{E0}$, all the encoder layers and the decoder are style-dependent layers.

As shown in Table 3, the multi-style AED ($m.L_{E6}$) can achieve on-par performance with the separately trained single-style model ($s.R$ and $s.L$), but the overall model footprint and run-time cost is reduced due to the shared style-agnostic encoder layers. When we increase the number of shared encoder layers to $E1 \sim E12$, we observe small negligible accuracy performance degradation in $m.L_{E12}$ comparing to $m.L_{E6}$. This suggests that we can largely leverage the shared style-agnostic bottom encoder layers when modeling multi-style transcription to save cost.

To illustrate the knowledge transfer behaviour of this architecture, we train a new multi-style AED ($m.L^{+}_{E6}$) with additional 50k hours data with lexical transcription (100k hours data with lexical transcription in total) and the same 50k hours with readable transcription. As shown in Table 3, the additional 50k hours data with lexical transcription not only helps in improving the performance of the lexical transcription branch, but also improves the readable branch. This confirms the knowledge transfer through the shared bottom layers in this architecture.

Table 3. Performance of the proposed multi-style AEDs with style-dependent layer ($m.L$), mixed-style output ($m.M$), and style-dependent prompt ($m.P$). E_i is the starting layer of the style-dependent encoder layer. for example, in $E6$, the bottom encoder layers $E1 \sim E6$ are style agnostic layers and top encoder layers $E7 \sim E18$ plus the decoder are style dependent layers.

	A			B		
Model	TER	F1	WER	TER	F1	WER
$m.L_{E6}$	16.6	0.71	6.2	28.5	0.60	9.0
$m.L_{E12}$	16.7	0.71	6.2	29.3	0.60	9.0
$m.L^{+}_{E6}$	15.9	0.72	5.5	27.0	0.62	8.3
$m.M0$	18.9	0.67	8.2	31.7	0.55	10.7
$m.M1$	19.0	0.66	6.7	29.0	0.57	9.4
$m.M2$	18.5	0.66	6.5	28.8	0.58	9.3
$m.P_{E6}$	21.7	0.68	7.2	31.9	0.57	9.6

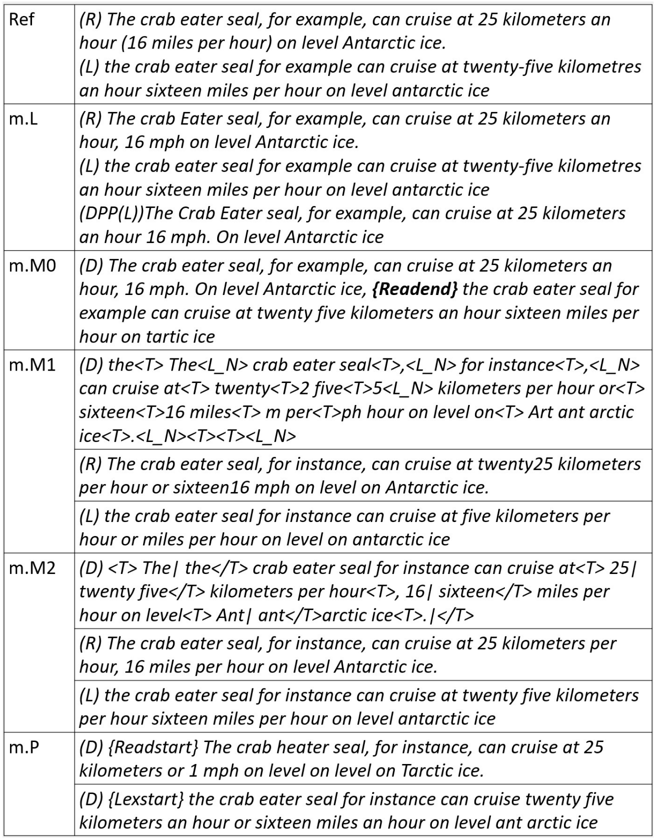
3.4. Result of Multi-Style AED with Style-Mixed Output

For the mixed-style AEDs with mixed-style output, we experimented with three different ways to compose the mixed-style output. $m.M0$ is the concatenated mixed-style AED described in Eq. 1, $m.M1$ and $m.M2$ are the two interlaced mixed-style AEDs described in Eq. 2 and Eq. 4. We found all three can generate reasonably good quality readable and lexical transcription within one decoding pass. The interlaced mixed-style AED with style alternative sessions ($m.M2$) performs the best. It achieves nearly on-par performance with the single-style AEDs especially on the conversation task (B). On the dictation task (A), the gap is slightly larger.

In the mixed-style AEDs with mixed-style output, the model needs to predict both the lexical/readable hypothesis and the tags introduced in the mixed-style language. The predicted tags should ideally follow the designated protocol so that the mixed-style transcription can be properly decoded to form different style transcription. We found the tag prediction is highly accurate, with occasional mistakes in complicated readable and lexical mixed-style cases. To address this, we introduced some simple mechanisms which allow the style parser to skip the illegitimate tags in case of mistakes in tag prediction and resume a normal style parsing. We can see some examples shown in Fig. 3.

3.5. Result of Multi-Style AED with Style-Dependent Prompt

For the prompt-based AED ($m.P$), the result seems to be not as promising as the other approaches. After looking into the details,



Ref	*(R) The crab eater seal, for example, can cruise at 25 kilometers an hour (16 miles per hour) on level Antarctic ice.* *(L) the crab eater seal for example can cruise at twenty-five kilometres an hour sixteen miles per hour on level antarctic ice*
m.L	*(R) The crab Eater seal, for example, can cruise at 25 kilometers an hour, 16 mph on level Antarctic ice.* *(L) the crab eater seal for example can cruise at twenty-five kilometres an hour sixteen miles per hour on level antarctic ice* *(DPP(L))The Crab Eater seal, for example, can cruise at 25 kilometers an hour 16 mph. On level Antarctic ice*
m.M0	*(D) The crab eater seal, for example, can cruise at 25 kilometers an hour, 16 mph. On level Antarctic ice,* ***{Readend}*** *the crab eater seal for example can cruise at twenty five kilometers an hour sixteen miles per hour on tartic ice*
m.M1	*(D) the<T> The<L_N> crab eater seal<T>,<L_N> for instance<T>,<L_N> can cruise at<T> twenty<T>2 five<T>5<L_N> kilometers per hour or<T> sixteen<T>16 miles<T> m per<T>ph hour on level on<T> Art ant arctic ice<T>.<L_N><T><T><L_N>*
	(R) The crab eater seal, for instance, can cruise at twenty25 kilometers per hour or sixteen16 mph on level on Antarctic ice.
	(L) the crab eater seal for instance can cruise at five kilometers per hour or miles per hour on level on antarctic ice
m.M2	*(D) <T> The| the</T> crab eater seal for instance can cruise at<T> 25| twenty five</T> kilometers per hour<T>, 16| sixteen</T> miles per hour on level<T> Ant| ant</T>arctic ice<T>.|</T>*
	(R) The crab eater seal, for instance, can cruise at 25 kilometers per hour, 16 miles per hour on level Antarctic ice.
	(L) the crab eater seal for instance can cruise at twenty five kilometers per hour sixteen miles per hour on level antarctic ice
m.P	*(D) {Readstart} The crab heater seal, for instance, can cruise at 25 kilometers or 1 mph on level on level on Tarctic ice.*
	(D) {Lexstart} the crab eater seal for instance can cruise twenty five kilometers an hour or sixteen miles an hour on level ant arctic ice

Fig. 3. Example of decoding results of the proposed multi-style AEDs. (D) refers to the raw decoding results, (R) and (L) refer to the extracted readable and lexical transcription.

we found this model can proceed decoding with proper readable or lexical transcription following the prompt, but it has a tendency to hallucinate. Hallucination largely brings down the accuracy score for the system, which would otherwise score comparably with the other proposed approach disregarding the hallucination part.

We plan to experiment with adding one-hot based style-embedding as in Fig. 2 (c) in addition to prompt in our future work to possibly address the hallucination issue.

3.6. Discussion

Fig. 3 presents some real decoding examples of the proposed multi-style AEDs. The readable and lexical style human transcription are also provided as reference.

For the AED with style-dependent layer, we observe the readable branch output has better segmentation comparing to applying DPP to the lexical hypothesis from the lexical branch. This is consistent with our hypothesis that leveraging acoustic information can help in segmentation comparing to using textual information alone.

For the AED with mixed-style output, we found that most of the time, the system can successfully predict the tags following the exact protocol used to compose the multi-style transcription during training. However, when it fails, it can create an issue in extracting the style-specific transcription. Fig. 3 shows such an example. For $m.M1$, $\langle T\rangle$ must be followed by one readable token and one lexical token including $\langle R_N\rangle$ and $\langle L_N\rangle$ for empty position. Noticeably, in this example, the system failed to predict $\langle R_N\rangle$ in two locations, which breaks the protocol and causes issues in properly decoding numbers. Although the wordpieces for both readable and lexical transcription are mostly correctly predicted, the style parser failed in placing them in the right style, which results in transcription errors. In comparison, $m.M2$ explicitly defines the starting and ending of a section with different transcription using $\langle T\rangle$ and $\langle/T\rangle$. Inside the section, different style transcription are separated by $|$. This design covers longer span which can avoid the order mismatch between the readable and lexical transcription. The above issue in $m.M1$ is resolved in $m.M2$. It is to be noted that the tag prediction is generally highly accurate. As mentioned before, we also introduced simple mechanisms to ignore the tag mistakes and resume the parsing from the next mixed-style segment.

For the AED with style-dependent prompt, we observe some segments are repeated, which is a kind of hallucination. Although it is hard to identify the exact causes of hallucination, we believe one contributing factor could be the complicated model sharing of different transcription styles in this architecture. Adding on-hot based style embedding can potentially alleviate this issue.

4. CONCLUSIONS

We presented a framework to generate multi-style transcription in an AED using three different architectures: (A) style-dependent layers; (B) mixed-style output; (C) style-dependent prompt. In this framework, multi-style transcription can be generated simultaneously or separately, through a single or multiple decoding passes on-demand. We show that the proposed framework can achieve nearly on-par performance compared to the single-style AED. Moreover, it provides an efficient data sharing mechanism across styles through knowledge transfer. We provided comprehensive comparison of the three approaches and pointed out the practical design consideration when choosing a specific architecture.

5. ACKNOWLEDGEMENT

The authors would like to acknowledge Yao Qian, Yu Shi, Vadim Mazalov, and Ruizhi Li for the helpful discussion.

6. REFERENCES

[1] K. H. Albrow, “The English writing system: Notes towards a description, journal schools council program in linguistics and english teaching papers series 2,” 1972.

[2] S. Peitz, M. Freitag, A. Mauser, and H. Ney, “Modeling punctuation prediction as machine translation,” in *Proceedings of the International Workshop on Spoken Language Translation (IWSL)*, 2011.

[3] Y. Yang Zhang, E. Bakhturina, K. Gorman, and B. Ginsburg, “NeMo inverse text normalization: From development to production,” in *https://doi.org/10.48550/arXiv.2104.05055*, 2022.

[4] R. Sproat and N. Jaitly, “RNN approaches to text normalization: A challenge,,” in *Proceedings of the International Workshop on Spoken Language Translation (IWSL)*, 2016.

[5] E. Pusateri, Bharat Ram Ambati, Elizabeth Brooks, Ondrej Platek, Donald McAllaster, and Venki Nagesha, “A mostly data-driven approach to inverse text normalization,” in *Proc. Interspeech 2017*, 2017, pp. 2784–2788.

[6] Sharman Tan, Piyush Behre, Nick Kibre, Issac Alphonso, and Shuangyu Chang, "Four-in-one: A joint approach to inverse text normalization, punctuation, capitalization, and disfluency for automatic speech recognition," in *2022 IEEE Spoken Language Technology Workshop (SLT)*, 2023, pp. 677–684.

[7] Piyush Behre, Sharman Tan, Padma Varadharajan, and Shuangyu Chang, "Streaming punctuation: A novel punctuation technique leveraging bidirectional context for continuous speech recognition," *International Journal on Natural Language Computing*, vol. 11, no. 6, pp. 01–13, dec 2022.

[8] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M Ostendorf, and M. Harper, "Enriching speech recognition with automatic detection of sentence boundaries and disfluencies," *Proceedings of IEEE Transactions on audio, speech, and language processing*, 2006.

[9] M. Sunkara, C. Shivade, S. Bodapati, and K. Kirchhoff, "Neural inverse text normalization," in *Proceedings of ICASSP 2021*, 2021.

[10] J. Liao, Y. Shi, M. Gong, L. Shou, S. Eskimez, L. Lu, H. Qu, and M. Zeng, "Improving readability for automatic speech recognition transcription," in *Proceedings of ICASSP*, 2021.

[11] E. Shriberg, A. Stolcke, D. HakkaniTur, and G. Tur, "Prosody-based automatic segmentation of speech into sentences and topics," *Speech communication*, vol. 32, no. 1-2, pp. 127–154, 2000.

[12] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," in *Proceedings of Interspeech*, 2017.

[13] A. Kim, T. Hori, and Watanabe. S., "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proceedings of ICASSP*, 2017.

[14] C. C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, K. Kannan, R. J. Weiss, Rao K., and K. Goninaetal, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proceedings of ICASSP*, 2018.

[15] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer," in *Proceedings of ASRU*, 2017.

[16] Y. He, T. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, and et al., "Streaming end-to-end speech recognition for mobile devices," in *ICASSP*, 2019.

[17] J. Li, R. Zhao, H. Hu, and Y. Gong, "Improving RNN transducer modeling for end-to-end speech recognition," in *Proceedings of ASRU*, 2019.

[18] K. Hu, T. Sainath, R. Pang, and R. Prabhavalkar, "Deliberation model based two-pass end-to-end speech recognition," in *Proceedings of ICASSP*, 2020.

[19] J. Li, "Recent advances in end-to-end automatic speech recognition," *APSIPA Transactions on Signal and Information Processing*, April 2022.

[20] P. K. O'Neill, V. Lavrukhin, S. Majumdar, V. Noroozi, Y. Zhang, O. Kuchaiev, J. Balam, Y. Dovzhenko, K. Freyberg, M.D. Shulman, B. Ginsburg, D. Watanabe, and G. Kucsko, "SPGIspeech: 5,000 hours of transcribed financial audio for fully formatted end-to-end speech recognition," in *Proceedings of Interspeech*, 2021.

[21] A. Radford, J. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *https://doi.org/10.48550/arXiv.2212.04356*, 2022.

[22] W. Kang, X. Yang, Z. Yao, F. Kuang, Y. Yang, L. Guo, L. Lin, and D. Povey, "ESB: A benchmark for multi-domain end-to-end speech recognition," in *https://arxiv.org/pdf/2210.13352.pdf*, 2022.

[23] S. Gandhi, P. V. Platen, and A. M. Rush, "LIBRIHEAVY: A 50,000 hours ASR corpus with punctuation casing and context," in *https://arxiv.org/pdf/2309.08105.pdf*, 2023.

[24] J. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proceedings of ICASSP*, 2013.

[25] Y. Huang, D. Yu, C. Liu, and Y. Gong, "Multi-accent deep neural network acoustic model with accent-specific top layer using the KLD-regularized model adaptation," in *Proceedings of Insterspeech*, 2014.

[26] M. Sperber and M. Paulik, "Speech translation and the end-to-end promise: Taking stock of where we are," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, July 2020, pp. 7409–7421, Association for Computational Linguistics.

[27] M. Sperber, H. Setiawan, C. Gollan, U. Nallasamy, and M. Paulik, "Consistent transcription and translation of speech," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 695–709, 2020.

[28] Orion Weller, Matthias Sperber, Christian Gollan, and Joris Kluivers, "Streaming models for joint speech recognition and translation," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online, Apr. 2021, pp. 2533–2539, Association for Computational Linguistics.

[29] T. Tanaka, R. Masumura, M. Ihori, A. Takashima, S. Orihashi, and N. Makishima, "End-to-end rich transcription-style automatic speech recognition with semi-supervised learning," in *Proceedings of Interspeech*, 2021.

[30] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: First results," in *arXiv preprint arXiv:1412.1602, 2014*, 2014.

[31] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proceedings of ICASSP*, 2017.

[32] S. Kim, T. Hori, and S. S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proceedings of ICASSP*, 2017.