INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Deliberation-Based Multi-Pass Speech Synthesis

Qingyun Dou, Xixin Wu, Moquan Wan, Yiting Lu, Mark J. F. Gales

University of Cambridge, United Kingdom
{qd212,xw369,mw545,ylt28,mjfg100}@cam.ac.uk

Abstract

Sequence-to-sequence (seq2seq) models have achieved state-of-the-art performance in a wide range of tasks including Neural Machine Translation (NMT) and Text-To-Speech (TTS). These models are usually trained with teacher forcing, where the reference back-history is used to predict the next token. This makes training efficient, but limits performance, because during inference the free-running back-history must be used. To address this problem, deliberation-based multi-pass seq2seq has been used in NMT. Here the output sequence is generated in multiple passes, each one conditioned on the initial input and the free-running output of the previous pass. This paper investigates, and compares, deliberation-based multi-pass seq2seq for TTS and NMT. For NMT the simplest form of multi-pass approaches, where the free-running first-pass output is combined with the initial input, improves performance. However, applying this scheme to TTS is challenging: the multi-pass model tends to converge to the standard single-pass model, ignoring the previous output. To tackle this issue, a guided attention loss is added, enabling the system to make more extensive use of the free-running output. Experimental results confirm the above analysis and demonstrate that the proposed TTS model outperforms a strong baseline.

Index Terms: sequence-to-sequence, speech synthesis

1. Introduction

Auto-regressive sequence-to-sequence (seq2seq) models with attention mechanisms are used in a variety of areas including Neural Machine Translation (NMT) [1, 2], Automatic Speech Recognition (ASR) [3] and speech synthesis [4, 5], also known as Text-To-Speech (TTS). These models excel at connecting sequences of different length, but can be difficult to train. A standard approach is teacher forcing, which guides a model with reference output history during training. This makes the model unlikely to recover from its mistakes during inference, where the reference output is replaced by generated output. This issue is often referred to as exposure bias. Several approaches have been introduced to tackle this issue, namely scheduled sampling [6], professor forcing [7] and attention forcing [8]. These approaches require sequential generation during training, and cannot be directly applied when parallel training is a priority.

Deliberation-based multi-pass seq2seq is a parallelizable alternative. It has been used in NMT [9] and ASR [10]. Here the output sequence is generated in multiple passes, each one conditioned on the initial input and the previous free-running output; the parameters for all passes are jointly optimized. This paper investigates, and compares, multi-pass TTS and NMT. A simpler form of multi-pass seq2seq is proposed: the free-running first-pass output is combined with the initial input, and the two passes are trained separately to enable parallel training. This scheme improves performance for NMT. However, applying it to TTS is challenging: the multi-pass model tends to converge to the standard single-pass model, ignoring the previous output. To tackle this issue, a guided attention loss is added, enabling the system to make more extensive use of the free-running output. NMT and TTS experiments confirm the above analysis.

2. Single-pass seq2seq

Sequence-to-sequence generation can be defined as mapping an input sequence $\boldsymbol{x}_{1:L}$ to an output sequence $\boldsymbol{y}_{1:T}$. From a probabilistic perspective, a model $\boldsymbol{\theta}$ estimates the distribution of $\boldsymbol{y}_{1:T}$ given $\boldsymbol{x}_{1:L}$, typically as a product of conditional distributions: $p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:L};\boldsymbol{\theta}) = \prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{x}_{1:L};\boldsymbol{\theta})$.

Ideally, the model is trained through minimizing the KL-divergence between the true distribution $p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:L})$ and the estimated distribution. In practice, this is approximated by minimizing the Negative Log-Likelihood (NLL) over some training data $\{\boldsymbol{y}_{1:T}^{(n)}, \boldsymbol{x}_{1:L}^{(n)}\}_1^N$, sampled from the true distribution:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:L} \sim p(\boldsymbol{x}_{1:L})} \mathrm{KL}\big(p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:L}) || p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:L};\boldsymbol{\theta})\big) \quad (1)$$

$$\propto -\sum_{n=1}^{N} \log p(\boldsymbol{y}_{1:T}^{(n)}|\boldsymbol{x}_{1:L}^{(n)};\boldsymbol{\theta}) \quad (2)$$

$\mathcal{L}(\boldsymbol{\theta})$ denotes the loss. During inference, given an input $\boldsymbol{x}_{1:L}^*$, the output can be obtained through searching for the most probable sequence from $p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:L}^*;\boldsymbol{\theta})$. The exact search is expensive and is often approximated by greedy search for continuous output, or beam search for discrete output [6].

2.1. Attention based models

Attention mechanisms [11, 12] are commonly used to connect sequences of different length. This paper focuses on attention-based encoder-decoder models. For these models:

$$\begin{aligned} p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{x}_{1:L};\boldsymbol{\theta}) &\approx p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\alpha}_t, \boldsymbol{x}_{1:L};\boldsymbol{\theta}) \\ &\approx p(\boldsymbol{y}_t|\boldsymbol{s}_t, \boldsymbol{c}_t;\boldsymbol{\theta}_y) \end{aligned} \quad (3)$$

$\boldsymbol{\theta} = \{\boldsymbol{\theta}_y, \boldsymbol{\theta}_s, \boldsymbol{\theta}_\alpha, \boldsymbol{\theta}_h\}$. $\boldsymbol{\alpha}_t$ is an alignment vector (a set of attention weights). $\boldsymbol{s}_t$ is a state vector representing the output history $\boldsymbol{y}_{1:t-1}$, and $\boldsymbol{c}_t$ is a context vector summarizing $\boldsymbol{x}_{1:L}$. Figure 1 and the following equations give a more detailed illustration of how $\boldsymbol{\alpha}_t$, $\boldsymbol{s}_t$ and $\boldsymbol{c}_t$ can be computed:

$$\boldsymbol{h}_{1:L} = f(\boldsymbol{x}_{1:L};\boldsymbol{\theta}_h) \quad (4)$$

$$\boldsymbol{s}_t = f(\boldsymbol{y}_{1:t-1};\boldsymbol{\theta}_s) \quad (5)$$

$$\boldsymbol{\alpha}_t = f(\boldsymbol{s}_t, \boldsymbol{h}_{1:L};\boldsymbol{\theta}_\alpha) \quad \boldsymbol{c}_t = \textstyle\sum_{l=1}^{L} \alpha_{t,l} \boldsymbol{h}_l \quad (6)$$

$$\hat{\boldsymbol{y}}_t \sim p(\cdot|\boldsymbol{s}_t, \boldsymbol{c}_t;\boldsymbol{\theta}_y) \quad (7)$$

The encoder maps $\boldsymbol{x}_{1:L}$ to $\boldsymbol{h}_{1:L}$ in context. For each decoder step, $\boldsymbol{s}_t$ summarizes $\boldsymbol{y}_{1:t-1}$. With $\boldsymbol{h}_{1:L}$ and $\boldsymbol{s}_t$, the attention mechanism computes $\boldsymbol{\alpha}_t$, and then $\boldsymbol{c}_t$. Finally, the decoder estimates a distribution based on $\boldsymbol{s}_t$ and $\boldsymbol{c}_t$, and optionally generates an output token $\hat{\boldsymbol{y}}_t$. Note that while illustrated with this form of attention, the ideas in this paper are not limited to it.

http://dx.doi.org/10.21437/Interspeech.2021-1405

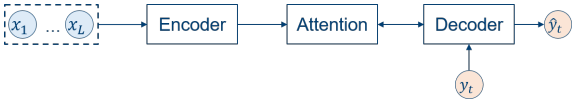

Figure 1: *Illustration of an attention-based encoder-decoder*

2.2. Training approaches

Equations 1 and 2 motivate teacher forcing, where the reference output history is given to the model, and the loss is:

$$\mathcal{L}(\boldsymbol{\theta}) = -\sum_{t=1}^{T} \log p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{x}_{1:L}; \boldsymbol{\theta}) \quad (8)$$

From this section on, the sum over the training set is omitted for the simplicity of description. This approach yields the correct model (zero KL-divergence) if: 1) the model is powerful enough ; 2) the model is optimized correctly; 3) there is enough training data to approximate the expectation in equation 1. However, these assumptions are often not true, hence the model is prone to mistakes that can accumulate across time.

In practice, the model is often assessed by some distance $\mathcal{D}$ between the reference $\boldsymbol{y}_{1:T}$ and the prediction $\hat{\boldsymbol{y}}_{1:T}$. This motivates Minimun Bayes Risk (MBR) training, which minimizes the expectation of $\mathcal{D}(\boldsymbol{y}_{1:T}, \hat{\boldsymbol{y}}_{1:T})$. This approach allows directly optimizing $\mathcal{D}$ [13, 14]. $\mathcal{D}$ does not need to be differentiable, and $\boldsymbol{y}_{1:T}$ and $\hat{\boldsymbol{y}}_{1:T}$ do not need to be aligned. However, for many tasks such as TTS, there is no gold-standard distance metric, and the alignment can be essential.

Although defined for sequences, $\mathcal{D}$ is usually computed at sub-sequence level, e.g. BLEU score for NMT and L_p distance for TTS. So training the model to predict the reference output, based on erroneous output history, indirectly reduces the Bayes risk. One example is to train the model in free running mode, where the generated output history is used, and the probability term in equation 8 becomes $p(\boldsymbol{y}_t|\hat{\boldsymbol{y}}_{1:t-1}, \boldsymbol{x}_{1:L}; \boldsymbol{\theta})$. This approach often struggles to converge, and several approaches are proposed to tackle this problem, namely scheduled sampling, professor forcing, and attention forcing.

Scheduled sampling [6] randomly decides whether the reference or generated output is added to the history. The probability term in equation 8 becomes $p(\boldsymbol{y}_t|\widetilde{\boldsymbol{y}}_{1:t-1}, \boldsymbol{x}_{1:L}; \boldsymbol{\theta})$; $\widetilde{\boldsymbol{y}}_t = \boldsymbol{y}_t$ with probability ϵ, and $\hat{\boldsymbol{y}}_t$ otherwise. ϵ gradually decays from 1 to 0 with a heuristic schedule. For professor forcing [7], the model outputs two sequences for each input sequence, respectively in teacher forcing mode and free running mode. The output and/or some hidden sequences are used to train a discriminator, which estimates the probability that a group of sequences is generated in teacher forcing mode. For the generator, there are two training objectives: 1) the standard NLL loss; 2) to fool the discriminator. Attention forcing [8] guides the model with the generated output history and reference attention. Here the probability term in equation 8 becomes $p(\boldsymbol{y}_t|\hat{\boldsymbol{y}}_{1:t-1}, \boldsymbol{\alpha}_t, \boldsymbol{x}_{1:L}; \boldsymbol{\theta})$. A problem with the above approaches is that they require sequential generation during training, and cannot be directly applied when parallel training is a priority. To our knowledge, teacher forcing is the most standard training approach for TTS, especially when neural vocoders are used or parallel training is important. The frame rate is often reduced for teacher forcing, not the model, even though it results in noisier waveform [8].

3. Deliberation-based multi-pass seq2seq

The main idea of deliberation-based multi-pass seq2seq is to generate the output in multiple passes, each one conditioned on the initial input and the previous free-running output. As the model learns to correct the free-running output, it alleviates the exposure bias. Without loss of generality, this section describes a two-pass seq2seq system, shown in figure 2. For this system:

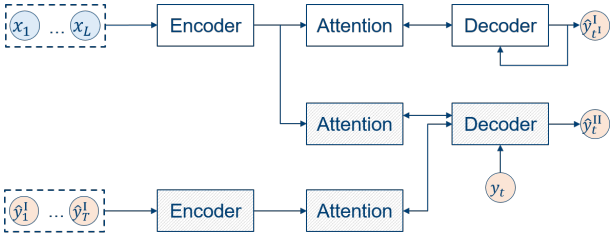

Figure 2: *Illustration of deliberation-based multi-pass seq2seq*

$$p(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:L}; \boldsymbol{\theta}^{\mathrm{I}}, \boldsymbol{\theta}^{\mathrm{II}}) = \sum_{\boldsymbol{y}^{\mathrm{I}}_{1:T^{\mathrm{I}}}} p(\boldsymbol{y}^{\mathrm{I}}_{1:T^{\mathrm{I}}}|\boldsymbol{x}_{1:L}; \boldsymbol{\theta}^{\mathrm{I}}) p(\boldsymbol{y}_{1:T}|\boldsymbol{y}^{\mathrm{I}}_{1:T^{\mathrm{I}}}, \boldsymbol{x}_{1:L}; \boldsymbol{\theta}^{\mathrm{II}}) \quad (9)$$

$p(\boldsymbol{y}^{\mathrm{I}}_{1:T^{\mathrm{I}}}|\boldsymbol{x}_{1:L}; \boldsymbol{\theta}^{\mathrm{I}})$ is computed by a standard single-pass model $\boldsymbol{\theta}^{\mathrm{I}}$. $p(\boldsymbol{y}_{1:T}|\boldsymbol{y}^{\mathrm{I}}_{1:T^{\mathrm{I}}}, \boldsymbol{x}_{1:L}; \boldsymbol{\theta}^{\mathrm{II}})$ is computed by a model $\boldsymbol{\theta}^{\mathrm{II}}$ with an additional attention mechanism over the first-pass output. At time t, $\boldsymbol{\theta}^{\mathrm{II}}$ operates as follows.

$$\boldsymbol{s}_t = f(\boldsymbol{y}_{1:t-1}; \boldsymbol{\theta}^{\mathrm{II}}_s) \quad (10)$$

$$\boldsymbol{h}_{x,1:L} = f(\boldsymbol{x}_{1:L}; \boldsymbol{\theta}^{\mathrm{II}}_{h,x}) \quad (11)$$

$$\boldsymbol{h}_{y,1:T^{\mathrm{I}}} = f(\boldsymbol{y}^{\mathrm{I}}_{1:T^{\mathrm{I}}}; \boldsymbol{\theta}^{\mathrm{II}}_{h,y}) \quad (12)$$

$$\boldsymbol{\alpha}_{x,t} = f(\boldsymbol{s}_t, \boldsymbol{h}_{x,1:L}; \boldsymbol{\theta}^{\mathrm{II}}_{\alpha,x}) \quad \boldsymbol{c}_{x,t} = \sum_{l=1}^{L} \alpha_{x,t,l} \boldsymbol{h}_{x,l} \quad (13)$$

$$\boldsymbol{\alpha}_{y,t} = f(\boldsymbol{s}_t, \boldsymbol{h}_{y,1:T^{\mathrm{I}}}; \boldsymbol{\theta}^{\mathrm{II}}_{\alpha,y}) \quad \boldsymbol{c}_{y,t} = \sum_{l=1}^{T^{\mathrm{I}}} \alpha_{y,t,l} \boldsymbol{h}_{y,l} \quad (14)$$

$$\hat{\boldsymbol{y}}^{\mathrm{II}}_t \sim p(\cdot|\boldsymbol{s}_t, \boldsymbol{c}_{x,t}, \boldsymbol{c}_{y,t}; \boldsymbol{\theta}^{\mathrm{II}}_y) \quad (15)$$

$\boldsymbol{\theta}^{\mathrm{II}}$ is based on $\boldsymbol{\theta}^{\mathrm{I}}$. It has two encoder-attention pairs, one for the initial input $\boldsymbol{x}_{1:L}$, another for the first-pass output $\boldsymbol{y}^{\mathrm{I}}_{1:T^{\mathrm{I}}}$. The encoder for $\boldsymbol{x}_{1:L}$ share the same parameters as that of $\boldsymbol{\theta}^{\mathrm{I}}$, i.e. $\boldsymbol{\theta}^{\mathrm{II}}_{h,x} = \boldsymbol{\theta}^{\mathrm{I}}_h$. The state vector $\boldsymbol{s}_t$ tracking the back-history is used by both attention mechanisms. The probability of $\boldsymbol{y}_t$ depends on $\boldsymbol{s}_t$, $\boldsymbol{c}_{x,t}$ and $\boldsymbol{c}_{y,t}$. In this paper, $\boldsymbol{c}_{x,t}$ and $\boldsymbol{c}_{y,t}$ are concatenated to form a new context vector. The intermediate output of $\boldsymbol{\theta}^{\mathrm{I}}$, such as $\boldsymbol{s}^{\mathrm{I}}_{1:T^{\mathrm{I}}}$ and $\boldsymbol{c}^{\mathrm{I}}_{1:T^{\mathrm{I}}}$, can optionally be combined with the $\boldsymbol{y}^{\mathrm{I}}_{1:T^{\mathrm{I}}}$ as the input to $\boldsymbol{\theta}^{\mathrm{II}}_{h,y}$. This extra connection requires more parameters, but speeds up training (in our TTS experiments). During inference, equation 10 will become $\boldsymbol{s}_t = f(\hat{\boldsymbol{y}}^{\mathrm{II}}_{1:t-1}; \boldsymbol{\theta}^{\mathrm{II}}_s)$.

It is difficult to jointly train $\boldsymbol{\theta}^{\mathrm{I}}$ and $\boldsymbol{\theta}^{\mathrm{II}}$ by maximizing the likelihood in equation 9. The expectation over $\boldsymbol{y}^{\mathrm{I}}_{1:T^{\mathrm{I}}}$ is intractable, and is often approximated by a sampling approach [9]. This paper investigates a simpler scheme and train the two models separately. First $\boldsymbol{\theta}^{\mathrm{I}}$ is trained with teacher forcing, as shown in equation 8. Then the expectation is approximated by a point estimate. Only one free-running output $\hat{\boldsymbol{y}}^{\mathrm{I}}_{1:T^{\mathrm{I}}}$ is generated from $\boldsymbol{\theta}^{\mathrm{I}}$, and $\boldsymbol{\theta}^{\mathrm{II}}$ is again trained with teacher forcing:

$$\mathcal{L}_y(\boldsymbol{\theta}^{\mathrm{II}}) = -\log p(\boldsymbol{y}_{1:T}|\hat{\boldsymbol{y}}^{\mathrm{I}}_{1:T^{\mathrm{I}}}, \boldsymbol{x}_{1:L}; \boldsymbol{\theta}^{\mathrm{II}}) \quad (16)$$

3.1. Machine translation and speech synthesis

For NMT, the input and output are both text sequences. This makes the application relatively simple. The additional encoder $\boldsymbol{\theta}^{\mathrm{II}}_{h,y}$ and attention mechanism $\boldsymbol{\theta}^{\mathrm{II}}_{\alpha,y}$ can be exactly the same as $\boldsymbol{\theta}^{\mathrm{II}}_{h,x}$ and $\boldsymbol{\theta}^{\mathrm{II}}_{\alpha,x}$, as both the initial input $\boldsymbol{x}_{1:L}$ and the first-pass output $\boldsymbol{y}^{\mathrm{I}}_{1:T^{\mathrm{I}}}$ are discrete text sequences.

For TTS, $\boldsymbol{x}_{1:L}$ is a discrete text sequence, and $\boldsymbol{y}^{\text{I}}_{1:T^{\text{I}}}$ is a continuous speech sequence. In most cases $\boldsymbol{y}^{\text{I}}_{1:T^{\text{I}}}$ is a feature sequence, and a neural vocoder maps it to a waveform. The difference between the text and feature input spaces means that the additional encoder needs to be modified. In this work, the text embedding layer in $\boldsymbol{\theta}^{\text{II}}_{h,x}$ is replaced by the linear layer in $\boldsymbol{\theta}^{\text{II}}_{h,y}$. $\boldsymbol{x}_{1:L}$ and $\boldsymbol{y}^{\text{I}}_{1:T^{\text{I}}}$ are also very different in length. Longer sequences are harder for the attention mechanism, and reduction in time resolution alleviates the problem. For example, pyramid encoder is often used in attention-based ASR models [3]. In this paper, adjacent frames in $\boldsymbol{y}^{\text{I}}_{1:T^{\text{I}}}$ are stacked in groups of four, before being fed into $\boldsymbol{\theta}^{\text{II}}_{h,y}$.

Our initial experiments show that the above techniques are not sufficient for performance gain in TTS. Very often, the second-pass model converged to the standard single-pass model, ignoring the free-running output. To tackle this issue, a guided attention loss [15], shown in equation 17, is added. This loss encourages the attention $\boldsymbol{\alpha}_{y,1:T}$ to be diagonal, enabling $\boldsymbol{\theta}^{\text{II}}$ to make more extensive use of the free-running output. For $\boldsymbol{\theta}^{\text{II}}$, the complete loss is $\mathcal{L}_y(\boldsymbol{\theta}^{\text{II}}) + \gamma\mathcal{L}_\alpha(\boldsymbol{\theta}^{\text{II}})$; γ is a scaling factor and g is a hyper-parameter controling the sharpness.

$$\begin{aligned} \mathcal{L}_\alpha(\boldsymbol{\theta}^{\text{II}}) &= \textstyle\sum_{t=1}^{T}\sum_{l=1}^{T^{\text{I}}}[\alpha_{y,t,l}w_{t,l}] \\ w_{t,l} &= 1 - \exp\left(-(t/T - l/T^{\text{I}})/2g^2\right) \end{aligned} \tag{17}$$

In this paper, Laplace distribution is assumed for TTS, and $\mathcal{L}_y \propto \sum_{t=1}^{T}||\boldsymbol{y}_t - \hat{\boldsymbol{y}}^{\text{II}}_t||_1$. When $\mathcal{L}_\alpha$ is used, it is important to monitor $\boldsymbol{\alpha}_{y,1:T}$ and the inference performance on a validation set via objective metrics such as Global Variance (GV). When $\boldsymbol{\alpha}_{y,1:T}$ is sharply diagonal, $\mathcal{L}_\alpha$ is low, but GV may degrade. Hence early stopping is essential.

3.2. Related work

In a similar fashion to training approaches such as scheduled sampling and attention forcing, multi-pass seq2seq addresses the problems of teacher forcing. An advantage of the proposed approach is that sequential generation is unnecessary during training. As the two passes are separately trained with teacher forcing, non-recurrent models such as Transformer [16] can be trained in parallel. If needed, the sequential generation of $\hat{\boldsymbol{y}}^{\text{I}}_{1:T^{\text{I}}}$ can be done beforehand. On the other hand, if training speed is not an issue, the multi-pass models can be combined with other training approaches to achieve even better performance.

Deliberation-based multi-pass seq2seq have been successful in NMT [9] and ASR [10]. This work focuses on TTS, where the application is more challenging due to speech sequences being long and strongly correlated in time. For NMT and ASR, the additional attention maps one text sequence to another, where copying an input token is often a reasonable option [17]. In contrast, for TTS the additional attention connects two speech sequences. Copying is usually not a good option, and finding the right focus is much harder. In terms of training, this work investigates a simple but effective scheme. For TTS, the infinite number of possible outputs makes MBR less practical than in NMT and ASR. So unlike references [9] and [10], this work does not train with MBR. This avoids sequential sampling during training, and improves efficiency. Some recent works on TTS [18, 19, 20, 21] use a duration model instead of attention. Multi-pass seq2seq is compatible with such models, as the extra attention connecting two speech sequences can be added in a similar way as described in section 3.1.

Table 1: *BLEU of various NMT systems*

model	BLEU↑
TF	31.10$\pm$0.27
SS	31.45$\pm$0.45
AF	31.54$\pm$0.14
FR-TF	**31.74**$\pm$0.27
TF-TF	31.29$\pm$0.05

4. Experiments

4.1. Machine translation

Data The experiments are conducted with the English-to-French data in IWSLT [22] 2015. The training set contains 208k sentence pairs. The validation (tst2013) and test (tst2014) sets respectively contain 1026 and 1305 sentence pairs. All the sentences are transcriptions or translations of TED talks.

Model, Training and Inference The single-pass baseline model is similar to Google's RNN-based model [23]. The differences are: the encoder has 2 layers of BLSTM and the decoder has 4 layers of LSTM; Luong attention [24] is used; the word embeddings have 200 dimensions. This model is trained with Teacher Forcing (TF) for 60 epochs. Starting from the baseline, two stronger single-pass models are fine-tuned with sequence-level Scheduled Sampling (SS) or Attention Forcing (AF) for 30 epochs. For SS, the probability of using the reference output decreases linearly from 1 to 0.9; more aggressive schedules are found to be harmful. The AF settings follow [25]. For the multi-pass system, the first-pass model is the same as the baseline, trained with TF and then used to generate a Free-Running (FR) output. The second-pass model is as described in section 3.1. The dimension of the decoder's input layer is increased by 400 to take an extra context vector. The second-pass model is randomly initialized and trained with TF for 50 epochs. Adam optimiser is used with a learning rate of 0.002 and the maximum gradient norm is 1. The learning rate is halved during fine-tuning. The batch size is 50. Dropout is used with a probability of 0.2. The inference approach is beam search with beam size 1. Checkpoints are selected based on validation performance. The effective number of epochs is smaller than the maximum, i.e. training goes on until convergence.

Evaluation Metrics BLEU [26] is used to measure the overall translation quality. The average of 1-to-4 gram BLEU scores are computed and a brevity penalty is applied. As a common practice, each model is trained 5 times with different random seeds and the mean $\pm$ standard deviation is reported. The code for computing BLEU score is available online.[1]

Results and analysis Table 1 shows the BLEU scores of various models. TF, SS and AF denote single-pass models trained with different approaches. SS and AF outperforms TF, as they address the exposure bias. FR-TF denotes the proposed multi-pass system. It outperforms all of the above. To see if this results from fixing the errors in the FR output, instead of a bigger model, an extra experiment is run. TF-TF denotes this experiment, where the first-pass output is generated in TF instead of FR mode. TF-TF has the same number of parameters as FR-TF, but its BLEU score is considerably lower. This indicates that the performance gain of FR-TF mainly results from fixing the errors in the FR output.

An interesting finding was that for the second pass, the initial input seems to be more important than the previous FR out-

[1] `https://github.com/moses-smt/mosesdecoder`

put. In an additional experiment, the input text is masked out, and only the FR output is given to the second-pass model. The BLEU dropped to 29.31, even lower than the baseline. This indicates that mapping French text with errors to its clean version is more difficult than mapping clean English to clean French.

4.2. Speech synthesis

Data The TTS experiments are conducted on LJ dataset [27], which contains 13100 utterances (about 24 hours) from one speaker. The utterances last from 1s to 10s. The train-valid-test split is 12600-250-250. The sampling rate is 22050Hz. Data preprocessing is the same as Tacotron [28].

Model, Training and Inference The single-pass models and their training follow Tacotron described by Table 1 in [28], except that: the attention mechanism is location-based [12]; a learning rate schedule is adopted.[2] The baseline is trained with TF for 350k steps. Starting from the baseline, two stronger single-pass models are fine-tuned with SS or AF for 80k steps. Checkpoints are selected based on validation performance, and training further does not yield improvements. For SS, the probability of using the reference output decreases linearly from 1 to 0.8; decreasing further is found to be harmful. For AF, the scale of the attention loss is 1. For the multi-pass system, the first-pass model is the same as the baseline. The second-pass model is as described in section 3.1. Its additional encoder, attention mechanism, and the first layer of decoder are randomly initialized, and the rest are initialized with the baseline. The attention RNN state is concatenated with the first-pass output, forming the input to the additional encoder. For $\mathcal{L}_\alpha$, the scale γ is 10, and the sharpness coefficient g is 0.4. The second-pass model can be trained in less than 4k steps. The neural vocoder and its training follow WaveRNN [29]. During inference, all the models operate in free running mode.

Evaluation Metrics MOS tests and AB preference tests are conducted using AMTurk. Each type of test is taken by 54 workers in the US. In a MOS test, a worker rates the overall quality of 5 groups of audio samples on the 5-point scale; within a group, the same text is mapped to speech by different systems. In a AB preference test, a worker listens to 10 pairs of samples, and indicates which is better overall. Following ESPnet [30], for each worker, the samples are randomly selected from the first 100 test sentences. The MOS tests help benchmarking the overall performance. The AB preference tests show a more direct comparison. Objective metrics are used over all test sentences. The expressiveness of speech is measured by Global Variance (GV) [31] of the feature sequence. Dynamic Time Warpped (DTW) L_1 distance between the reference and the generated feature sequences is also computed. The distance is normalized by the length of the reference. Both GV and DTW are averaged over the test set and feature dimensions. High-quality samples should have low DTW distance and high GV.

Results and analysis Table 2 shows the MOS, GV and DTW L_1 distance of various TTS systems. In terms of MOS, SS is marginally better than TF. AF outperforms TF, which is consistent with reference [8]. The proposed multi-pass system (FR-TF) outperforms all single-pass systems. Figure 3 shows some more direct AB preference comparisons. It is clear that both AF and FR-TF outperform TF, and that FR-TF is slightly better than AF. The objective metrics show similar trends, meaning that they can be good indicators of human perception. In terms of expressiveness, AF is known to be better than TF [8]. FR-TF achieves slightly higher GV than AF, although the difference is less obvious in the samples. In general, we find FR-TF less expressive than AF, but more stable. For AF and FR-TF, the empirical frequency of attention failure is respectively 4% and 2%. Similar to the case of NMT, TF-TF is not nearly as good as FR-TF in GV and DTW, showing the the performance gain of FR-TF results from fixing errors of the FR output. Considering its objective performance, TF-TF is excluded in the MOS test.

[2]The code and some generated speech samples are available at `http://mi.eng.cam.ac.uk/~qd212/2021interspeech`

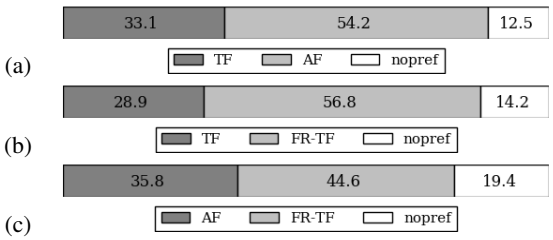

Figure 3: *AB preference tests comparing TF, AF and FR-TF*

Table 2: *MOS, GV and DTW distance of various TTS systems*

	MOS↑	GV↑	DTW↓
reference	$4.42_{\pm 0.09}$	0.0235	0
TF	$3.67_{\pm 0.11}$	0.0171	6.29
SS	$3.70_{\pm 0.12}$	0.0167	6.02
AF	$3.89_{\pm 0.10}$	0.0219	**5.59**
FR-TF	**4.03**$_{\pm 0.10}$	**0.0223**	5.64
TF-TF	—	0.0136	6.73

Similar to NMT, we run the experiment where the initial input is masked. Here the GV and DTW distance are 0.0130 and 7.46, much worse than the baseline. This shows that mapping FR speech to its reference is more difficult than mapping text to the reference speech, i.e. the initial input is more important for the second pass. This also explains why the second-pass model tends to ignore the FR output when no guided attention loss is used. Following [8], a TF baseline and a FR-TF system are trained at 200Hz, where the exposure bias is more severe. While the GV and DTW distance of TF degrade considerably to 0.012 and 7.42, those of FR-TF remain at a similar level (0.2117 and 5.768). This further demonstrates the effectiveness of FR-TF.

5. Conclusions

This paper investigates, and compares, deliberation-based multi-pass seq2seq for TTS and NMT. A parallelizable multi-pass scheme is proposed: the first-pass output is combined with the initial input, and the two passes are separately trained. This scheme improves NMT performance. However, applying it to TTS is challenging: the multi-pass model tends to converge to the standard single-pass model, ignoring the previous output. To tackle this issue, a guided attention loss is added, enabling the system to make more extensive use of the free-running output. Experimental results confirm the above analysis and demonstrate that the proposed TTS model outperforms a strong baseline. A natural line of future research is to apply multi-pass seq2seq to duration-based TTS models.

6. Acknowledgements

The authors would like to thank Kate Knill for helping with the experiments, and ALTA Institute, Cambridge Assessment, CSC and Cambridge Trust for financial support.

7. References

[1] G. Neubig, "Neural machine translation and sequence-to-sequence models: A tutorial," *arXiv preprint arXiv:1703.01619*, 2017.

[2] P.-Y. Huang, F. Liu, S.-R. Shiang, J. Oh, and C. Dyer, "Attention-based multimodal neural machine translation," in *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, 2016, pp. 639–645.

[3] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[4] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[5] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5180–5189.

[6] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179.

[7] A. M. Lamb, A. G. A. P. Goyal, Y. Zhang, S. Zhang, A. C. Courville, and Y. Bengio, "Professor forcing: A new algorithm for training recurrent networks," in *Advances In Neural Information Processing Systems*, 2016, pp. 4601–4609.

[8] Q. Dou, J. Efiong, and M. J. Gales, "Attention forcing for speech synthesis," *Proc. Interspeech 2020*, pp. 4014–4018, 2020.

[9] Y. Xia, F. Tian, L. Wu, J. Lin, T. Qin, N. Yu, and T.-Y. Liu, "Deliberation networks: Sequence generation beyond one-pass decoding," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 1782–1792.

[10] K. Hu, T. N. Sainath, R. Pang, and R. Prabhavalkar, "Deliberation model based two-pass end-to-end speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7799–7803.

[11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015.

[12] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems*, 2015, pp. 577–585.

[13] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," *Proc. ICLR*, 2016.

[14] D. Bahdanau, P. Brakel, K. Xu, A. Goyal, R. Lowe, J. Pineau, A. Courville, and Y. Bengio, "An actor-critic algorithm for sequence prediction," *Proc. ICLR*, 2017.

[15] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4784–4788.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6000–6010.

[17] M. X. Chen, O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, M. Schuster, N. Shazeer, N. Parmar *et al.*, "The best of both worlds: Combining recent advances in neural machine translation," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 76–86.

[18] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Advances in Neural Information Processing Systems*, 2019, pp. 3171–3180.

[19] Y. Ren, C. Hu, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text-to-speech," *ICLR 2021*, 2021.

[20] C. Yu, H. Lu, N. Hu, M. Yu, C. Weng, K. Xu, P. Liu, D. Tuo, S. Kang, G. Lei *et al.*, "Durian: Duration informed attention network for multimodal synthesis," *arXiv preprint arXiv:1909.01700*, 2019.

[21] J. Donahue, S. Dieleman, M. Bińkowski, E. Elsen, and K. Simonyan, "End-to-end adversarial text-to-speech," *Proc. ICLR*, 2021.

[22] M. Cettolo, C. Girardi, and M. Federico, "Wit3: Web inventory of transcribed and translated talks," in *Conference of european association for machine translation*, 2012, pp. 261–268.

[23] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[24] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.

[25] Q. Dou, Y. Lu, P. Manakul, X. Wu, and M. J. Gales, "Attention forcing for machine translation," *arXiv preprint arXiv:2104.01264*, 2021.

[26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[27] K. Ito, "The LJ speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[28] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *Proc. Interspeech 2017*, pp. 4006–4010, 2017.

[29] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2410–2419.

[30] T. Hayashi, R. Yamamoto, K. Inoue, T. Yoshimura, S. Watanabe, T. Toda, K. Takeda, Y. Zhang, and X. Tan, "ESPnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7654–7658.

[31] T. Nose and T. Kobayashi, "An intuitive style control technique in HMM-based expressive speech synthesis using subjective style intensity and multiple-regression global variance model," *Speech Communication*, vol. 55, no. 2, pp. 347–357, 2013.