Speak, Read and Prompt: High-Fidelity Text-to-Speech with Minimal Supervision

- Arxiv
- Google Research
- Paper, Demo

최예린
서강대학교 인공지능학과
Email: lakahaga@u.sogang.ac.kr
2023.4.14

Overview

- **Encoder-Decoder 스타일의 대규모 TTS 모델**
  - **Cf) VALL-E : GPT-3 like TTS**
- **Parallel data (audio-text paired data) 를 소량 사용하기 위한 방법 제안**
  - **Pretrain – backtranslation – finetuning을 거침**

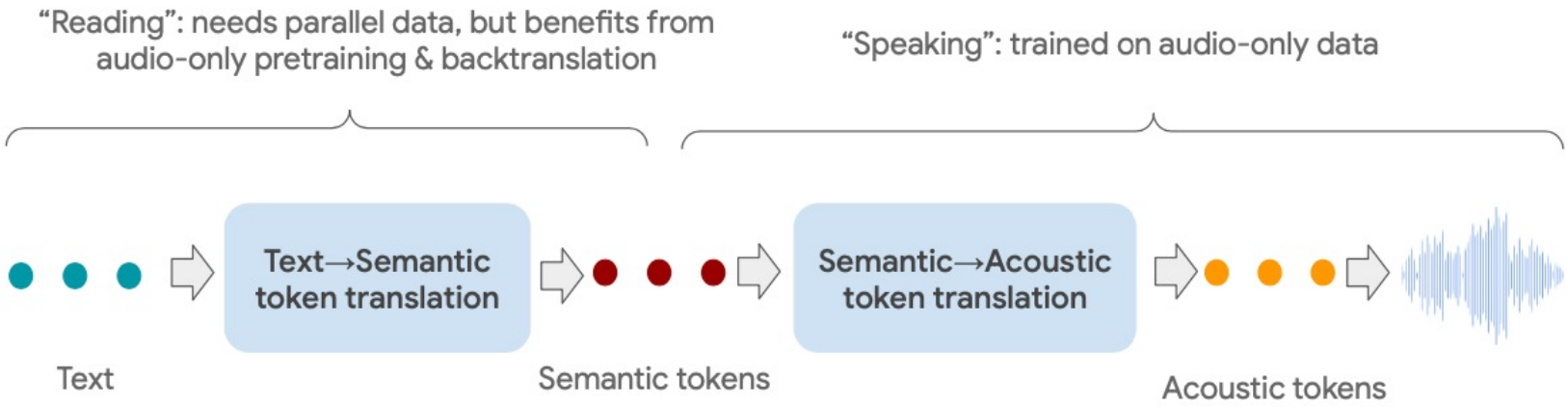


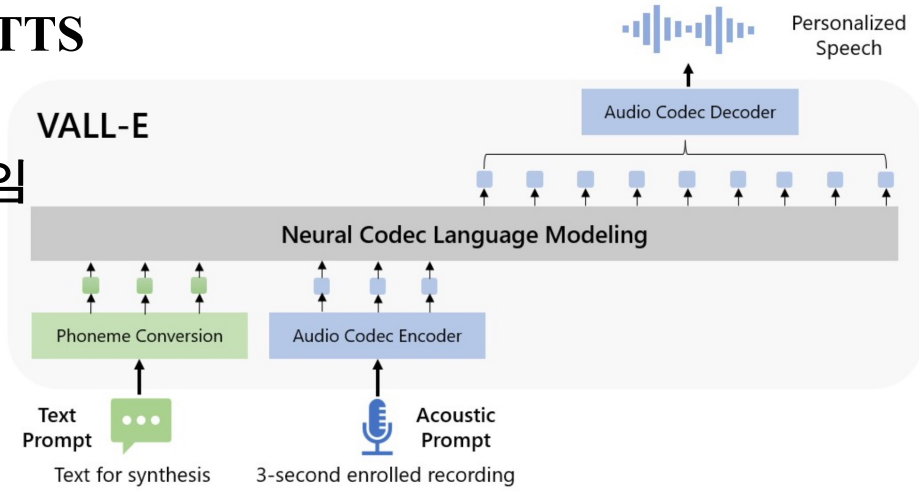
Figure 1: **SPEAR-TTS.** The first stage $\mathcal{S}_1$ ("reading") maps tokenized text to semantic tokens. The second stage $\mathcal{S}_2$ ("speaking") maps semantic tokens to acoustic tokens. Acoustic tokens are decoded to audio waveforms.

Remind – VALL-E

- *Neural Codec Language Models are Zero-shot Text to Speech Synthesis*
- **Audio codec decoder를 이용한 GPT-3 like TTS**
  - 훈련 데이터도 기존 600 시간 에서 60k시간으로 늘림 (LibriLight)
  - Noisy data여도 많은 양을 학습시키면 된다!
  - 학습 데이터의 규모를 키워 generalization 성능을 높임
    - 이때, transcript를 ASR 모델을 이용하여 구성

- **기존 TTS와 달리 audio codec을 이용(first in TTS)** (cf. mel-spectrogram)

- **Prompt-based approach 를 이용하여 zero-shot TTS**
  - **High speaker similarity with 3초의 speaker 음성**
- **TTS에서의 In-context learning capability를 보임**



SPEAR-TTS

- S_1: text input -> discrete semantic tokens
- S_2 : semantic tokens -> acoustic tokens
- Semantic token : w2v-BERT representation / acoustic tokens : audio codec code

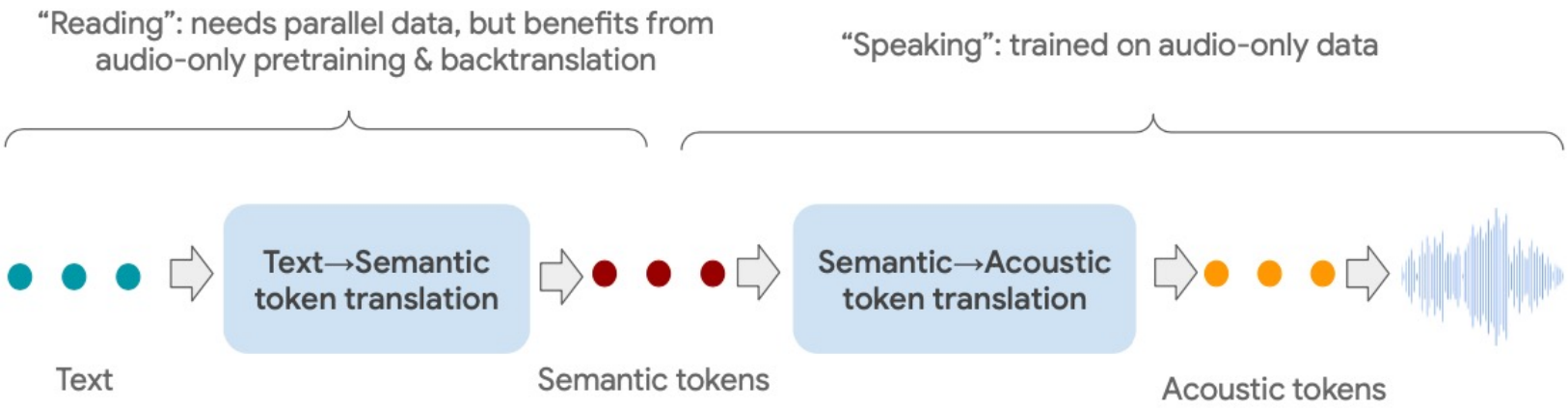
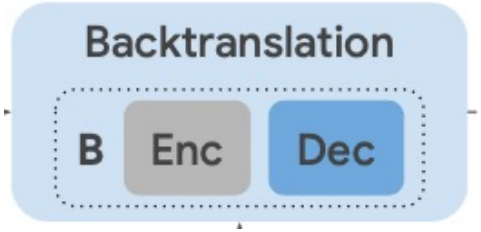


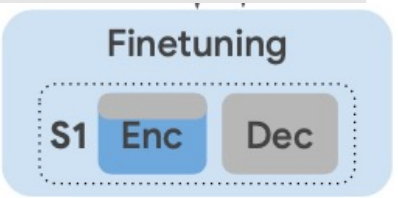
Figure 1: **SPEAR-TTS.** The first stage $\mathcal{S}_1$ ("reading") maps tokenized text to semantic tokens. The second stage $\mathcal{S}_2$ ("speaking") maps semantic tokens to acoustic tokens. Acoustic tokens are decoded to audio waveforms.

SPEAR-TTS - S_1

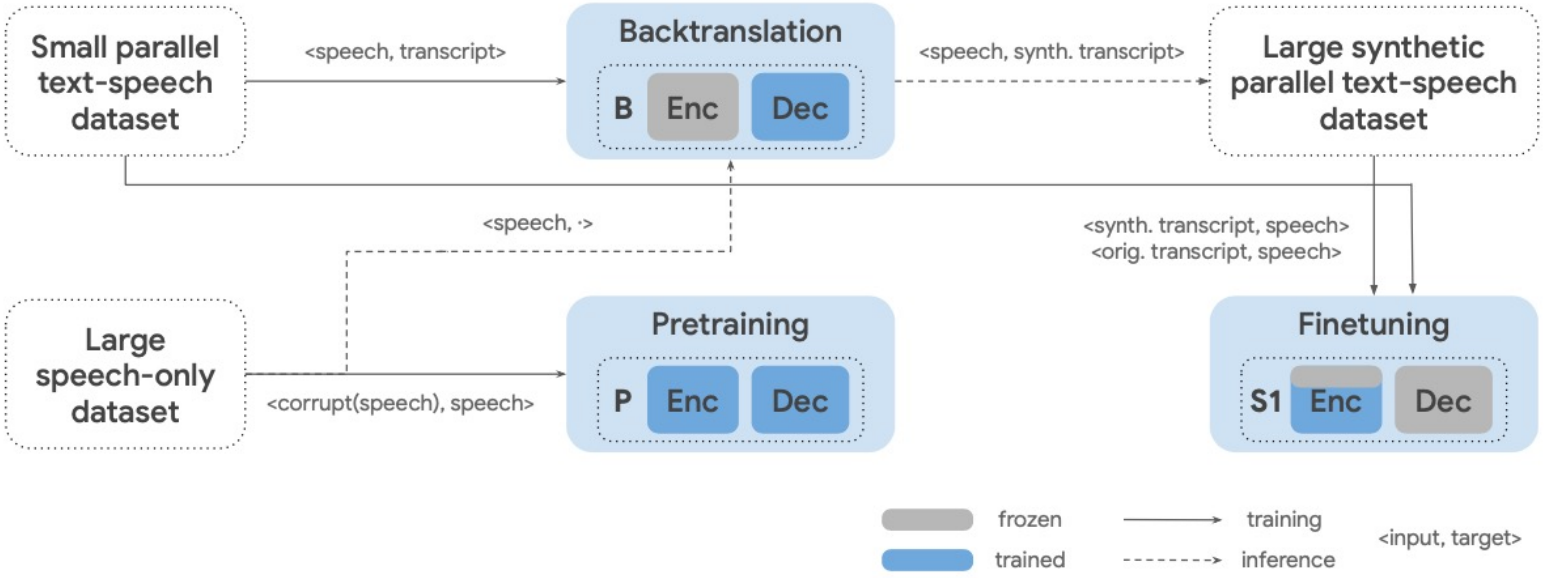
- Pretraining
  - Pre-text task: BART, T5 에서 쓰는 token/span denoise
  - Input: corrupted speech(semantic tokens) / output: speech(semantic tokens)
  - Corpus: LibriLight (600k hours) / w2v-bert representation
- Backtranslation
  - Asr 처럼 사용 – input: speech(semantic tokens) / output : transcript
  - Pretrained model에서 encoder는 Freeze하고, decoder만 Update
  - Audio만 있는 corpus에서 transcript를 생성
  - LibriTTS의 Transcript를 생성해서 사용 (551 hours)
    - Backtranslation에서 Transcript 생성 성능을 비교하고자



SPEAR-TTS - S_1



- **Finetuning**
  - **Input: text(transcript) / output: speech (semantic tokens)**
  - **Pretrained model에서 decoder 전체 + encoder의 upper layer를 freeze**
  - **Corpus: LJSpeech**
    - **Single speaker 로 훈련해도 multi-speaker에 inference 할 수 있다는 것을 보여줌**
    - **추가적으로, finetuning에 필요한 데이터 수량에 대한 실험도 진행**
- **S_1 전체 process**



SPEAR-TTS - S_2

- **Sampling**
  - 사용하는 이유
    - **audio only corpus는 background noise를 포함하는 경우가 있음**
    - **때문에 생성한 음성에도 noise가 포함될 수 있음**
    - **여러개 생성해서 가장 좋은 걸 써보자!**
  - **Method**
    - **Stochastic sampling**
    - **Multiple sequence for the same input**
    - **이때 최종 output을 고르는 기준은 비교 기준(reference audio)이 필요하지 않은 Metric**
      - DNSMOS 와 유사한 Metric이라고 논문에 표현
    - **3개를 생성하고 고름**

- ***DNSMOS: A Non-Intrusive Perceptual Objective Speech Quality Metric to Evaluate Noise Suppressors***
  - **CNN 모델로 MOS 점수 Prediction을 학습한 모델**

SPEAR-TTS - S_2

- Semantic tokens(w2v-bert) to acoustic tokens(audio codec discrete code)
  - Translation task로 seq2seq model 학습
- 12-layer decoder only Transformer
- 이때, speaker identity를 유지하기 위해 train 단계에서부터 prompt 형식 사용
  - Prompt : target speaker의 음성
- 훈련에서는 원음에서 겹치지 않는 segment 2개를 뽑아서
  - 하나는 prompt로, 하나는 target으로 사용
- Input
  - Semantic tokens from prompt
  - Semantic tokens from target
  - Acoustic tokens from prompt

  ⇒ 사이사이에 separate token을 추가

  ⇒ 기존 연구(VALL-E 이전)와 달리 prompt의 Text가 필요하지 않음
- Output
  - Acoustic tokens from target
- Autoregressive 하게 생성

Experiment

- 평가하고자 하는 것
  - **Backtranslation으로 생성한 transcript의 CER**
  - **Voice Diversity**
    - **Single speaker로 훈련한 모델이 다양한 speaker identity를 표현할 수 있는가?**
  - **Voice preservation**
    - **zero-shot speaker TTS에서 prompt로 주어진 speaker identity를 잘 표현하는가?**
  - **Overall quality (MOS)**
- 평가 데이터
  - **LibriSpeech test-clean**

- **Ablation Study**
  - **Pretraining이 진짜 필요한가?**
  - **Finetuning 에 필요한 최소한의 데이터 수량은?**

Result

1. **Backtranslation으로 생성한 transcript의 CER = 0.98**
   - **LibriSpeech test-clean에 대한 성능**
2. **Voice Diversity**
   - **Single speaker로 훈련한 모델이 다양한 speaker identity를 표현할 수 있는가?**
   - **Speaker classifier의 entropy를 측정**
     - **LibriSpeech train-clean-100, test-clean을 학습한 classifier**

Table 2: **Voice diversity (bits).** We measure the entropy of the empirical distribution of the voices detected by a speaker classifier.

	LJSpeech	LibriTTS		
# speakers	1	61	123	247
Ground-truth	2.55	5.82	6.71	7.68
SPEAR-TTS	6.11	6.22	6.16	6.28
FastSpeech2-LR	0.66	-	-	-

- **Speaker classifier 가 LJSpeech를 학습하지 않았음**
- **Classifier 로 detect한 speaker들의 entropy**
- **SPEAR-TTS는 single-speaker로 finetuning 했지만, 다양한 speaker표현이 가능함.**

Result

3. Voice Preservation

Table 3: **Voice preservation in prompted generation (classifier accuracy).** The $\mathcal{S}_1$ model is trained on 15 min of parallel data.

CER (%)	Speaker accuracy (%)		Voice diversity (bits)
	top-1	top-3	
1.92	92.4	98.1	0.41

Table 4: **Comparing voice preservation with baselines (cosine similarity).** Results for YourTTS and VALL-E are taken from (Wang et al., 2023, Table 2).

Model	Parallel training data	Cosine similarity
YourTTS	∼ 600 h	0.34
VALL-E	60,000 h	0.58
SPEAR-TTS	15 min	0.56

- **Table3: Speaker classifier 로 판단한 target speaker와의 유사성**
  - **Classifier 의 accuracy**
  - **Classifier 로 detect한 speaker들의 entropy**

- **Table 4: Cosine Similarity**
  - **Prompt의 acoustic token 과 합성음의 acoustic token의 cosine similarity**
  - **VALL-E에 비해 현저히 적은 parallel data로도 비슷한 유사성을 얻음**

Result

4. MOS

- **LJSpeech를 학습한 모델 비교**
  - 학습양이 현저히 적음에도 좋은 성능 (GT 보다도)

Table 5: **Mean Opinion Score (MOS) evaluation.** All compared systems are trained on subsets of LJSpeech (Ito and Johnson, 2017). ± indicates 95% CI obtained by bootstrap.

	FastSpeech2-LR			SPEAR-TTS	Ground-truth
Parallel training data	15 min	1 h	24 h	15 min	-
MOS	$1.72_{\pm 0.04}$	$2.08_{\pm 0.04}$	$2.11_{\pm 0.04}$	$\mathbf{4.96}_{\pm 0.02}$	$4.92_{\pm 0.04}$

- **Zero-shot speaker TTS 에 대한 성능 비교**

Table 6: **Mean Opinion Score (MOS) evaluation for prompted generation.** Prompts for both systems and samples for VALL-E are taken from the demo page of VALL-E. ± indicates 95% CI obtained by bootstrap.

System	VALL-E	SPEAR-TTS (15 min)
MOS	$3.35_{\pm 0.12}$	$\mathbf{4.75}_{\pm 0.06}$

- **VALL-E demo Page에 있는 음성과 성능 비교**

Result

- **Pretraining 없는 모델 / 학습 데이터량에 따른 성능 비교**

Table 1: **Intelligibility** of SPEAR-TTS and our baselines, depending on the training scenario and the amount of parallel data available from LJSpeech. We measure CER (%, lower is better) on LibriSpeech test-clean. ± indicates 95% CI obtained by bootstrap."×" indicates models that produce unintelligible speech.

		SPEAR-TTS			
		Training		Backtranslation	
Parallel training data	FastSpeech2-LR	from scratch (a)	Pretraining (b)	from scratch (c)	pretraining (d)
24 h	$1.99_{\pm 0.20}$	$3.67_{\pm 0.21}$	$2.38_{\pm 0.13}$	$2.26_{\pm 0.14}$	$2.06_{\pm 0.12}$
12 h	-	$4.31_{\pm 0.28}$	$2.54_{\pm 0.14}$	$2.27_{\pm 0.14}$	$2.03_{\pm 0.12}$
3 h	$2.52_{\pm 0.25}$	$20.1_{\pm 0.74}$	$3.07_{\pm 0.15}$	$2.21_{\pm 0.12}$	$2.01_{\pm 0.12}$
2 h	-	$24.7_{\pm 0.71}$	$3.73_{\pm 0.17}$	$2.22_{\pm 0.13}$	$2.09_{\pm 0.12}$
1 h	$2.74_{\pm 0.27}$	×	$5.51_{\pm 0.21}$	$2.23_{\pm 0.13}$	$2.16_{\pm 0.13}$
30 min	$3.18_{\pm 0.28}$	×	$21.3_{\pm 0.43}$	$2.52_{\pm 0.15}$	$2.20_{\pm 0.12}$
15 min	$4.90_{\pm 0.34}$	×	×	$2.88_{\pm 0.19}$	$2.21_{\pm 0.12}$

- **(a): Pretraining 없이 scratch부터 parallel data로 학습함**
- **(c) ,(d) : parallel data + backtranslation 한 transcript까지 학습**
- **From scratch / FastSpeech2: parallel data의 양이 적어질 수록 성능이 떨어짐**
- **(a) vs (b) : Pretraining 후 Finetuning한 모델은 2h 분량의 parallel data만 학습해도 성능 유지**
  - **scratch 부터 24h 학습한 성능 = 2h 분량 finetuning**
- **(a),(b) vs (c) ,(d) : audio-only data에서 backtranslation으로 뽑은 transcript까지 포함했을 때 성능이 좋고, Parallel data 양을 확 줄일 수 있음**

Conclusion

- 기존 VALL-E와 달리 text->semantic / semantic -> acoustic
  - Cf. VALL-E text+prompt -> acoustic

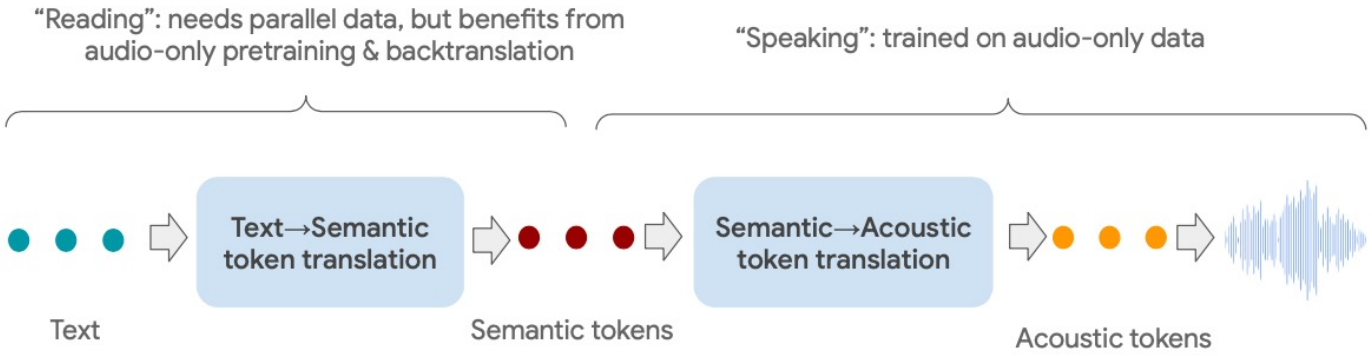


Figure 1: **SPEAR-TTS.** The first stage $\mathcal{S}_1$ ("reading") maps tokenized text to semantic tokens. The second stage $\mathcal{S}_2$ ("speaking") maps semantic tokens to acoustic tokens. Acoustic tokens are decoded to audio waveforms.

- Audio only data로 pretrain 후 소량의 Parallel(audio-text pair) 데이터로 finetuning으로 성능을 낼 수 있는 것을 확인
  - 특히, single speaker로만 Finetuning해도 다양한 speaker identity를 낼 수 있는 것을 확인

끝

감사합니다.