Incremental Text-to-Speech Synthesis Using Pseudo Lookahead with Large Pretrained Language Model

Paper link

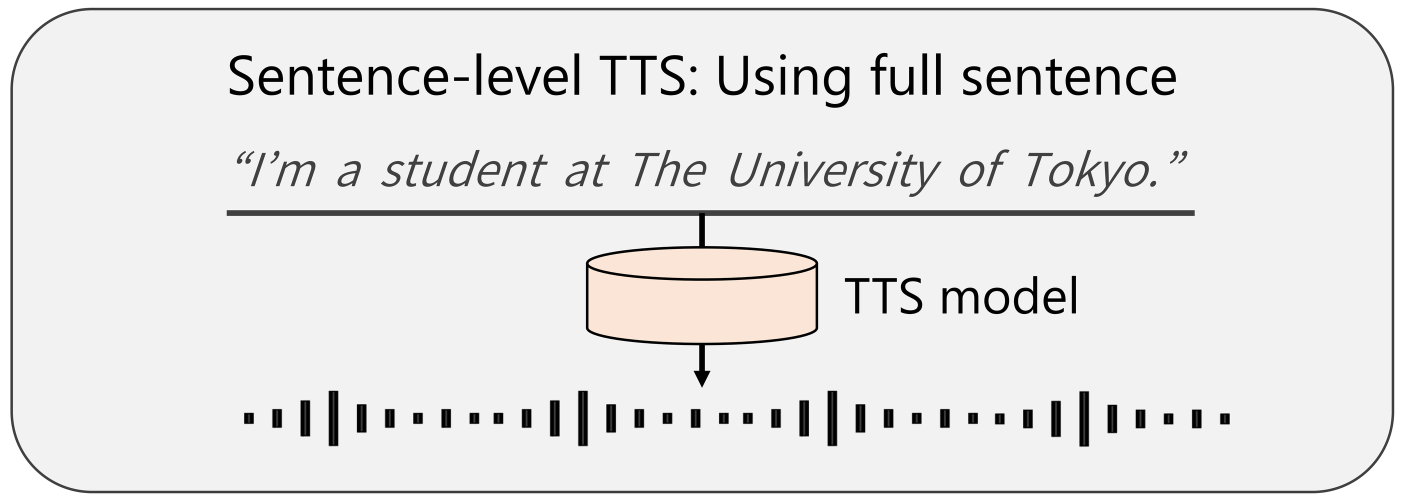
Takaaki Saeki, Shinnosuke Takamichi, Hiroshi Saruwatari
The University of Tokyo, Japan

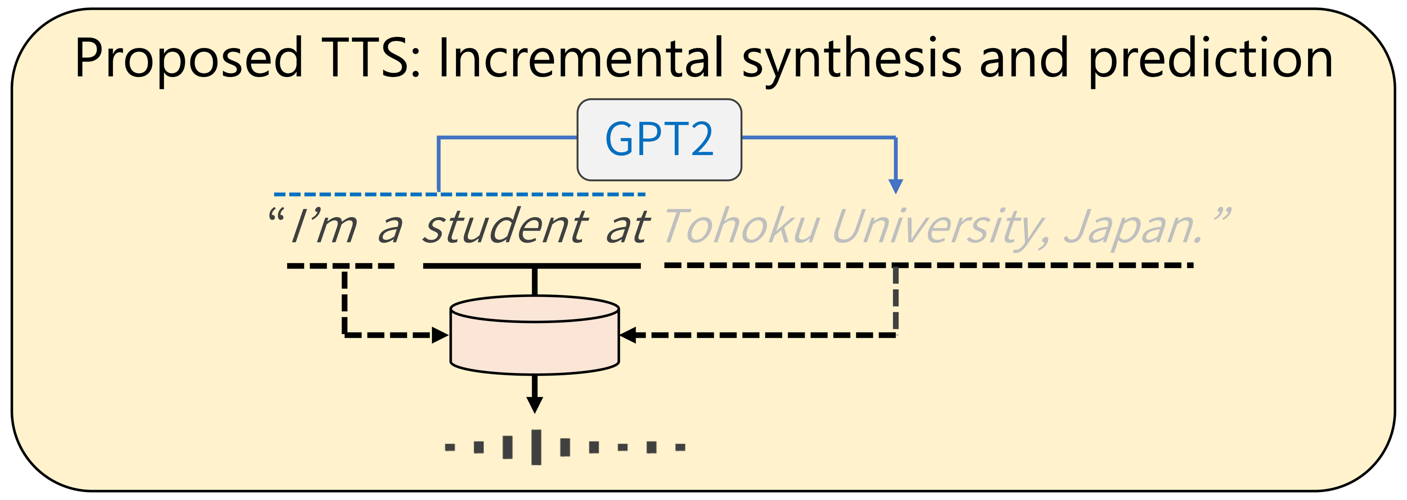
ICASSP 2022
SPE-39

- ❑ We propose **an incremental text-to-speech (TTS) synthesis method** that synthesizes speech in small linguistic units for streaming applications.
- ❑ Our method uses **pseudo lookahead** generated with a language model as **future context** to **address the tradeoff between naturalness and waiting time.**

Background: Incremental TTS for streaming application e.g., simultaneous speech translation)

- ❑ **Sentence-level TTS**: Using **full sentence** to generate output speech
- ❑ **Incremental TTS**: Synthesizing speech in **small linguistic units**
- ❑ Incremental TTS suffers from **tradeoff between quality and latency**
  - Synthesizing each segment independently [1] ⇨ Lower latency but lower quality
  - Waiting for k words (lookahead-*k policy*) [2] ⇨ Higher quality but need waiting time
- ❑ We propose an incremental TTS method **generated pseudo lookahead as future context**
  - Imitating human's incremental reading **(reading while predicting framework)**
  - ⇨ Achieving higher naturalness without waiting for future input words





Method

1) Incremental synthesis procedure

Generating pseudo lookahead with **GPT2** [3] from observed segment

⇩

Current segment, past observed segment, and pseudo lookahead are fed to encoder

⇩

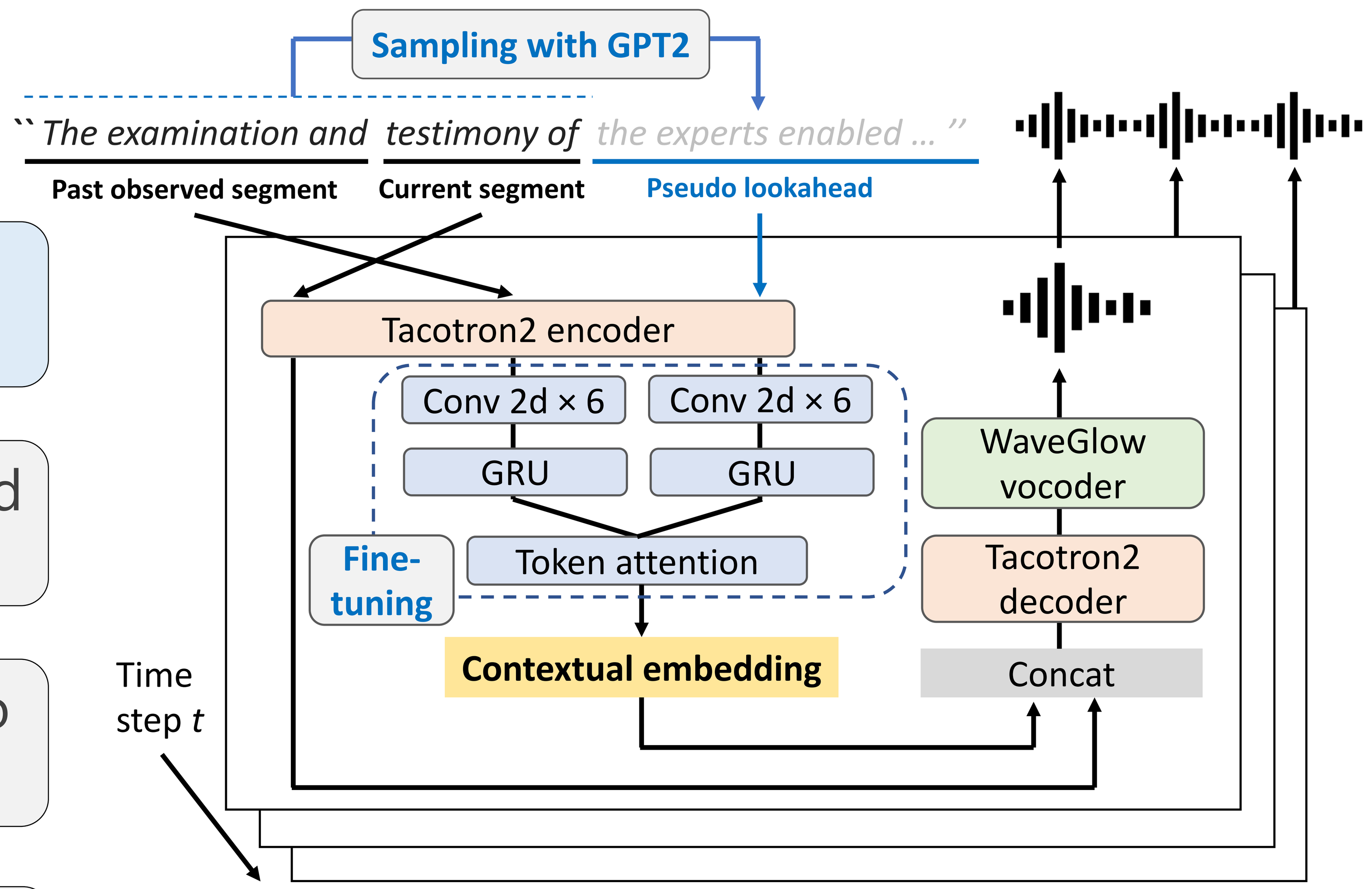
Encoded past observed segment and pseudo lookahead are fed to **contextual encoder**

⇩

Encoded current segment and contextual embedding are combined and fed to decoder

2) TTS model architecture and training

- ❑ Tacotron 2 [4]-based neural TTS model
- ❑ Training encoder, decoder, and context encoder in an **end-to-end manner**
- ❑ Training model with ground-truth lookahead



3) Language-model-guided fine-tuning

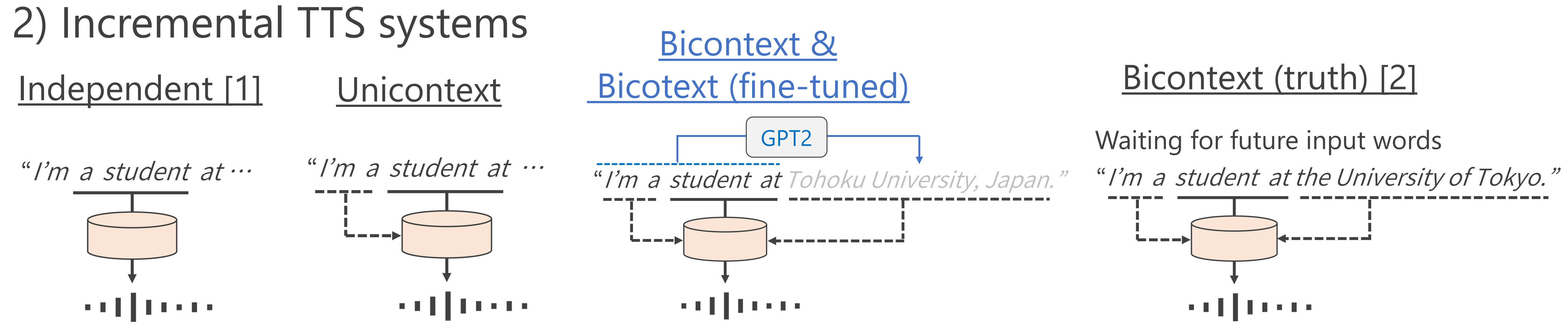
- ❑ **Fine-tuning only contextual encoder using language model** so that model better fits pseudo lookahead
- ❑ In addition to loss for TTS model training, we define **cosine similarity loss** between contextual embedding **with ground-truth lookahead** and **with pseudo lookahead**

Evaluation

1) Experimental conditions

Corpus	LJSpeech [5] (English, single speaker, 24h)
Number of words in each input segment	3 (training), 2 (inference)
Number of words in pseudo lookahead	5

2) Incremental TTS systems



3) Evaluation metrics

Objective evaluation: Calculated error rates with speech recognition model

Subjective evaluation: Mean opinion score (MOS) test with 40 native English speakers

Results of objective and subjective evaluations

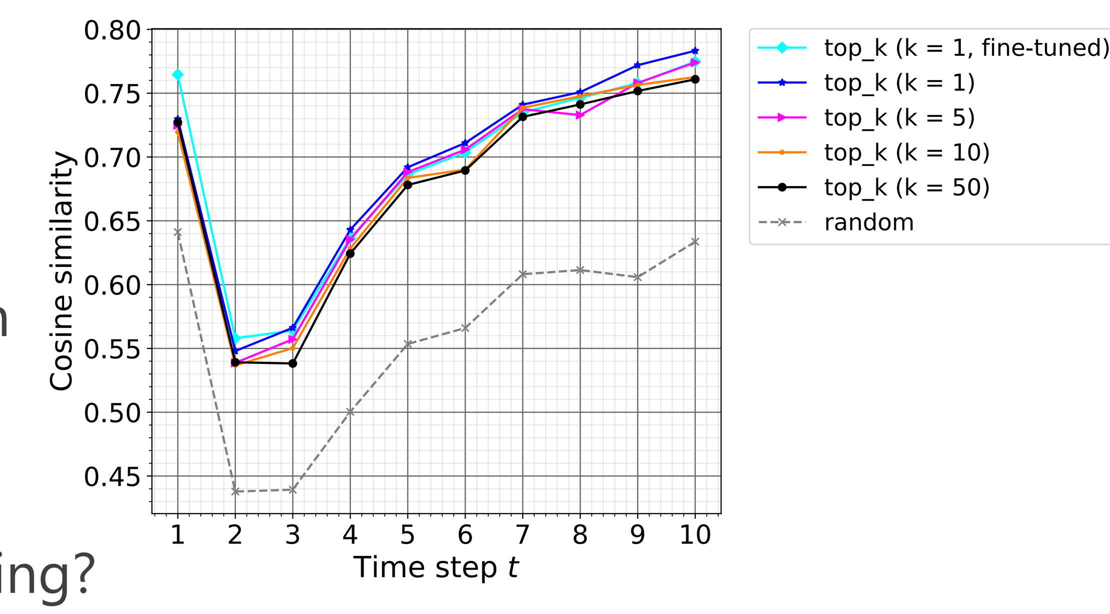
Method	CER	WER	MOS
Ground-truth	5.1 %	17.9 %	4.28 ± 0.13
Full-sentence	5.5 %	18.2 %	3.82 ± 0.12
Bicontext (truth)	8.2 %	24.2 %	3.36 ± 0.16
Independent	38.9 %	96.9 %	2.69 ± 0.20
Unicontext	22.8 %	53.9 %	2.99 ± 0.18
Bicontext	**11.9** %	**29.8** %	3.38 ± 0.14
Bicontext (fine-tuned)	**8.0** %	**22.5** %	3.44 ± 0.16

4) Results

Bicontext > Unicontext : Demonstrating the effectiveness of pseudo lookahead
Bicontext (fine-tuned) ≥ Bicontext (truth) : Equivalent to waiting for future words
Full-sentence > Bicontext (fine-tuned) : Need to further improve quality

5) Discussion

Cosine similarity between $\boldsymbol{e}_{\mathrm{truth}}$ and $\boldsymbol{e}_{\mathrm{pseudo}}$



$\boldsymbol{e}_{\mathrm{truth}}$: contextual emb. with ground-truth lookahead
$\boldsymbol{e}_{\mathrm{pseudo}}$: contextual emb. with pseudo lookahead

Q) Is precise word prediction needed?
⇨ GPT2 prediction is much better than random

Q) Random sampling or Maximum likelihood?
⇨ Maximum likelihood (k = 1) is better

Q) How fine-tuning affects contextual embedding?
⇨ Fine-tuning Improves contextual embedding in early time step (beginning of sentence)

Reference

[1] Yanagita et al., *Proc. SSW*, Sep. 2019.
[2] Ma et al., *Proc. EMNLP*, Nov. 2020.
[3] Radford et al., 2019.
[4] Shen et al., *Proc. ICASSP*, Apr. 2018.
[5] Ito et al., 2017.