大規模言語モデルによる未観測文の生成機構を持つ End-to-End インクリメンタル音声合成

佐伯 高明[1,a)]　高道 慎之介[1,b)]　猿渡 洋[1,c)]

概要：テキスト音声合成 (text-to-speech: TTS) は，テキスト情報から人間の発話音声を人工的に合成する技術である．近年の深層学習の発展に伴い，人間の自然音声と同程度に高品質な発話音声を生成できる end-to-end TTS モデルが提案されている．このような手法の多くは，発話文全体の長い時系列情報をモデルの入力として用いることで出力音声を合成する，発話文単位での TTS 手法である．しかし，同時音声翻訳への応用など，文が逐次的にしか観測されず，かつ低遅延な処理が必要な場合，小さな言語単位ごとに逐次的に音声合成を行うインクリメンタル TTS を確立する必要がある．一般に，インクリメンタル TTS では，出力音声品質と出力遅延との間にトレードオフが発生する．未観測の後続言語情報 (lookahead) を用いずに当該文セグメントからの合成を行う場合，遅延を限りなく抑えられる一方で，自然性の高い音声を出力することは困難である．反対に，未観測文の入力を待ってから生成処理を行うことで自然性は向上するが，後続文観測の待機に伴う出力遅延が発生する．本研究では，大規模言語モデルを用いて擬似 lookahead を生成することで，後続文の待機時間を発生させずに未観測のコンテキストを考慮するインクリメンタル TTS 手法を提案する．提案手法は，人間が文を逐次的に読み上げる際の文予測機能を，計算機的に模倣する手法と捉えることができ，多様なドメインのテキストデータで学習された GPT2 を用いて，汎用的な言語知識に基づく lookahead 生成を行う．実験的評価により，提案手法は，(1) 過去のコンテキストのみを考慮したインクリメンタル TTS 手法よりも有意に高品質な合成音声を出力でき，(2) 真の lookahead の観測を待つ場合と同程度の合成音声品質を達成できることを示す．

キーワード：インクリメンタル音声合成，end-to-end 音声合成，言語モデル，コンテキスト埋め込み

Takaaki Saeki[1,a)]　Shinnosuke Takamichi[1,b)]　Hiroshi Saruwatari[1,c)]

1. はじめに

同時音声翻訳 [1], [2] は，異言語間でのインタラクティブな音声コミュニケーションを可能にし，言葉の壁を取り去ることのできる技術である．この同時音声翻訳は，短時間の時系列情報を逐次的に処理する 3 つのモジュールからなり，それぞれ音声認識 (automatic speech recognition: ASR)・機械翻訳 (machine translation: MT)・テキスト音声合成 (text-to-speech: TTS) である．深層学習の発展に伴い，ASR や MT と同様に，TTS の品質と柔軟性は飛躍的に向上している．近年の TTS 手法は，深層学習モデルで発話文全体の時系列情報をモデル化することにより，人間の発話音声と同程度に高品質な発話音声を生成することが可能である．しかし，このような発話文単位での TTS 手法とは異なり，同時音声翻訳に用いるインクリメンタル TTS では，入力文が逐次的にしか観測されず，かつ発話文全体の入力を待つための遅延を許容できないため，数単語レベルの小さな言語単位を扱う必要がある．一般に，インクリメンタル TTS では，出力音声の自然性と合成にかかる遅延とのトレードオフが問題となる．低遅延なインクリメンタル TTS のためには，そのタイムステップでの当該文セグメントよりも未来の情報である未観測文 (lookahead) の入力を待たずに，既観測文のみを用いて当該文セグメントを処理することが理想的である．しかし，このような処理手法では，未観測のコンテキストに本来依存する当該音声セグメントを，そのコンテキストを無視して合成するた

[1] 東京大学 大学院情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo, Bunkyo, Tokyo 113–8656, Japan.
[a)] takaaki_saeki@ipc.i.u-tokyo.ac.jp
[b)] shinnosuke_takamichi@ipc.i.u-tokyo.ac.jp
[c)] hiroshi_saruwatari@ipc.i.u-tokyo.ac.jp

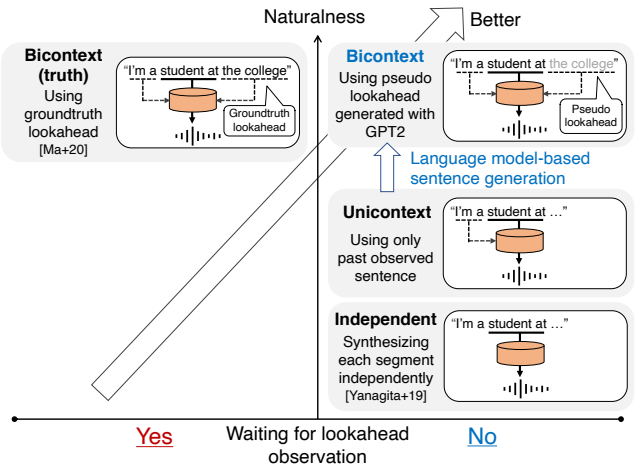

図 1 本研究の提案手法と比較手法の概要図.

め，合成音声の自然性が低くなる．反対に，後続の未観測文の入力を待ってから生成処理を行うことで，後続のコンテキストを考慮した上で当該音声セグメントを合成できるため，出力音声の自然性は向上するが，後続文の観測の待機に伴い出力遅延が大幅に増大する．

本稿では，大規模学習済み言語モデルにより生成した擬似 lookahead を用いて，高品質かつ低遅延な逐次生成を行うインクリメンタル TTS 手法を提案する．人間が文を逐次的に目視しながら読み上げを行う場合，既観測文から未観測文を予測し，前後のコンテキストに自然につながるように当該文セグメントを読み上げることができる．本研究の提案手法は，このような人間の逐次読み上げ機構を計算機的に模倣するために，多様なドメインのデータで学習された大規模言語モデルである GPT2 [3] を用いて lookahead を推定する．真の lookahead の入力を待つのではなく，GPT2 により生成した擬似 lookahead を未来のコンテキスト情報として用いることで，後続文の待機時間を発生させずに合成音声の自然性を改善できる．TTS モデルのアーキテクチャは Tacotron2 [4] に基づく encoder-decoder モデルであり，当該単語の前後の文脈を考慮するためのコンテキスト埋め込みネットワーク [5] を持つ．学習時に，encoder・decoder・コンテキスト埋め込みネットワークを一貫的に学習することで，当該単語を高品質に生成する．さらに，本研究では言語モデル誘導型学習による fine-tuning を提案する．これは，言語モデルによる擬似 lookahead 生成をコンテキスト埋め込みネットワークの学習過程に導入することで，GPT2 による生成文と真の lookahead の間のコンテキスト差異を提言する手法である．実験的評価により，提案手法は，(1) 既観測文のみを考慮した場合よりも有意に高品質な合成音声を出力でき，(2) 未観測文の待機時間を発生させずに，真の lookahead を待つ場合と同程度の出力音声品質を実現できることを示す．

本研究の提案手法および評価に用いた比較手法の概要を図 1 に示す．本研究の提案手法は，未観測文の入力を待たないため入力遅延を最小限に抑えることができ，かつ擬似 lookahead 生成により高い自然性を実現できる．各比較手法については 4.2 節で述べる．

2. 関連研究

近年，従来の統計的パラメトリック音声合成 [6], [7], [8] から，文字・音素列などを入力とする単一の深層学習モデルでメルスペクトログラム系列を直接推定する end-to-end TTS [4], [9], [10] への転換に伴い，音声合成の品質は飛躍的に向上している．そのような中，end-to-end TTS モデルを用いたインクリメンタル TTS 手法 [11], [12], [13], [14] が複数研究されている．この先駆けとして，Tacotron [9] を用いたインクリメンタル TTS 手法 [11] が提案されている．この手法は，本研究の提案手法と同様に，セグメントレベルで学習・生成を行う手法であるが，当該音声セグメントの生成時に前後のコンテキストを考慮しない点で異なる．この手法は，本稿の 4 節の “independent” と同様に当該音声セグメントを独立に生成するため，一般に出力音声の自然性は低くなる．Ma らは，機械翻訳の prefix-to-prefix フレームワークを用いて，毎タイムステップで k 単語だけ未来の情報を待って当該単語の生成を行う手法 (lookahead-k policy) [12] を提案している．この prefix-to-prefix フレームワークを用いた別の手法として，遅延と品質のトレードオフを最適にするために，強化学習を用いて扱うセグメントの単語数を動的に選択する手法 [14] も提案されている．本研究の提案手法は，これら 2 手法とは異なり，同時音声翻訳への応用に向けて未観測文の待機時間を発生させずに当該単語を即座に生成することを目的としている．さらに，本研究の TTS モデルでは，発話文単位の TTS の並列化に向けた手法 [5] で導入されているコンテキスト埋め込みネットワークを用いている．この手法は，イントネーションフレーズに着目することで，発話文内の各フレーズを並列に扱う手法であり，生成するフレーズの前後のフレーズが与えられている状況で生成処理を行う．それに対し，本研究で扱うインクリメンタル TTS では，当該文セグメントの後ろの言語情報は，そのタイムステップ時点では未観測である．

3. 手法

本節では，提案するインクリメンタル TTS 手法について述べる．3.1 節では，GPT2 による文章生成を統合した推論処理について述べる．3.2 節では，前後のコンテキストを考慮して当該音声セグメントの生成処理を行うためのモデル構造を説明し，3.3 節では言語モデル誘導型学習による fine-tuning について述べる．3.4 節では，提案手法について議論を行う．

3.1 言語モデルによる擬似 lookahead 生成

まず，TTS モデルの入力の**当該文セグメント**が N 個の単語を含んでいるとする．タイムステップ t での**既観測文**を $\boldsymbol{w}_{1:Nt} = \boldsymbol{w}_1, \cdots, \boldsymbol{w}_n, \cdots, \boldsymbol{w}_{Nt}$ とし，そのうち N 単語からなる末尾の単語列 $\boldsymbol{w}_{N(t-1)+1:Nt} = \boldsymbol{w}_{N(t-1)+1}, \cdots, \boldsymbol{w}_{Nt}$ を当該文セグメントとする．ここで，$\boldsymbol{w}_n$ は n 番目の単語である．このとき，既観測文のうち当該文セグメントを含まない単語列 $\boldsymbol{w}_{1:N(t-1)} = \boldsymbol{w}_1, \cdots, \boldsymbol{w}_{N(t-1)}$ を**過去の既観測文**として，既観測文と区別する．GPT2 [3] は自己回帰型の言語モデルであり，M 単語からなる単語列 $\boldsymbol{w}_{1:M}$ の確率分布を，以下のように条件付き確率分布の積に分解する．

$$p(\boldsymbol{w}_{1:M}) = \prod_{m=1}^{M} p\left(\boldsymbol{w}_m | \boldsymbol{w}_{1:m-1}\right) \tag{1}$$

このモデル化に基づき，確率分布 $p(\boldsymbol{w}_{Nt+1:Nt+L} | \boldsymbol{w}_{1:Nt})$ からサンプリングを行うことで，L 単語からなる未観測の単語列 $\hat{\boldsymbol{w}}_{Nt+1:Nt+L} = \hat{\boldsymbol{w}}_{Nt+1}, \cdots, \hat{\boldsymbol{w}}_{Nt+L}$ が得られる．この $\hat{\boldsymbol{w}}_{Nt+1:Nt+L}$ を**擬似 lookahead** と呼び，当該単語よりも未来のコンテキストを表す単語列として用いる．TTS モデルでは，単語 $\boldsymbol{w}_n$ を扱う際に文字または音素列を用いるため，単語 $\boldsymbol{w}_n$ に対応する文字または音素列を $\boldsymbol{x}_n$ と定義する．TTS モデルを $G(\cdot)$ と定義すると，出力すべきメルスペクトログラム $\boldsymbol{y}_t$ は以下のように得られる．

$$\boldsymbol{y}_t = G\left(\boldsymbol{x}_{N(t-1)+1:Nt} | \boldsymbol{x}_{1:N(t-1)}, \hat{\boldsymbol{x}}_{Nt+1:Nt+L}, \boldsymbol{\theta}_G\right) \tag{2}$$

ここで，$\boldsymbol{\theta}_G$ は $G(\cdot)$ のモデルパラメータである．$\boldsymbol{z}_t$ をメルスペクトログラム列 $\boldsymbol{y}_t$ から生成した音声波形とするとき，波形生成処理は WaveGlow [15] ボコーダ $V(\cdot)$ を用いて以下のように実現される．

$$\boldsymbol{z}_t = V\left(\boldsymbol{y}_t | \boldsymbol{\theta}_V\right) \tag{3}$$

ここで，$\boldsymbol{\theta}_V$ は $V(\cdot)$ のモデルパラメータである．この $\boldsymbol{z}_t$ を，過去のタイムステップまでに得られた音声波形 $\boldsymbol{z}_{1:t-1}$ に対して接続していくことで，逐次的に合成音声を出力する．

3.2 TTS モデル

本研究で用いるモデル構造は，Tacotron2 [4] に基づき，過去の既観測文と未観測文を考慮するコンテキスト埋め込みネットワーク [5] を統合した end-to-end TTS モデルである．モデル構造の詳細を図 2 に示す．まず，当該文セグメント・過去の既観測文・未観測文を Tacotron2 の encoder に入力し，過去の既観測文と未観測文に対応する encoder 出力をコンテキスト encoder に入力する．ここで，このコンテキスト encoder は 6 層の 2 次元畳み込み層と 1 層の gated recurrent unit (GRU) 層をスタックしたものである．2 つのコンテキスト encoder はパラメータを共有しており，モデルサイズなどのハイパーパラメータは Cong らの手法 [5] と同一のものを用いている．それぞれのコンテキスト encoder の出力を結合し，global style token [16] に基づく token attention 層に入力することで，コンテキスト埋め込みを得る．これら一連の，Tacotron2 の encoder 出力を入力とし，コンテキスト埋め込みを出力するネットワークを，“コンテキスト埋め込みネットワーク” と呼ぶ．得られたコンテキスト埋め込みと，当該文セグメントに対応する Tacotron2 encoder の出力を結合し，Tacotron2 decoder に入力する．Tacotron2 の encoder・decoder・コンテキスト埋め込みネットワークを一貫的に学習することで，前後の文脈を考慮しながら当該文セグメントのメルスペクトログラム列を生成することができる．

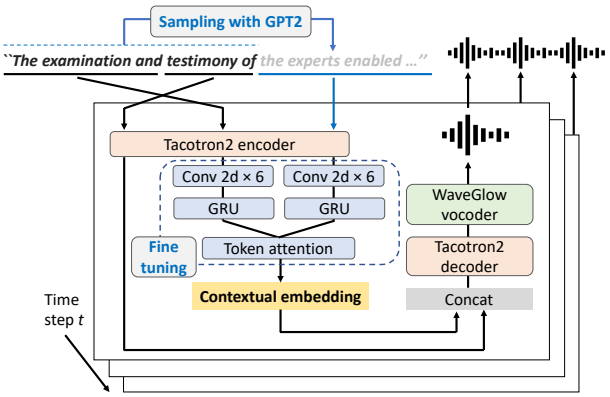

図 2 提案手法のモデルアーキテクチャ．

TTS モデルを学習する際の未観測文には，学習データに含まれる真の未観測文を用いる．学習のための前処理では，Cong らの手法 [5] と同様に，固定の窓長・ホップサイズを持つ sliding text window を発話文に適用することで，過去の既観測文・当該文セグメント・未観測文へと分割する．さらに，当該文セグメントに対応する波形を forced alignment によって抽出し，目標データとする．

3.3 言語モデル誘導型学習による fine-tuning

3.1 節で述べたように，擬似 lookahead 生成により，大規模言語モデルの言語知識をインクリメンタル TTS に対して用いることが可能である．しかし，この手法では，学習時に用いる真の lookahead と，推論時に用いる擬似 lookahead の間にミスマッチが生じる．つまり，TTS モデルの学習時に言語モデルによる lookahead 生成を考慮していないため，推論時に擬似 lookahead を未観測のコンテキストとして十分に活用できないことが考えられる．

したがって，本研究では，擬似 lookahead をインクリメンタル TTS に対してより効果的に用いるための言語モデル誘導型学習を提案する．3.2 節で述べた学習過程とは異なり，GPT2 によって生成した擬似 lookahead を未観測文として用いながら fine-tuning を行う．その際，3.2 節で述べた sliding text window で抽出した過去の既観測文と当該

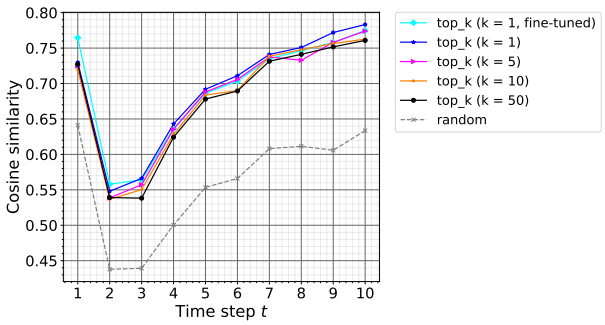

図 3 平均コサイン類似度.

文セグメントを GPT の入力とし，擬似 lookahead を生成して学習データとして用いる．$\boldsymbol{e}_{\text{pseudo}}$ を，擬似 lookahead を未観測文として用いた際のコンテキスト埋め込みとし，$\boldsymbol{e}_{\text{truth}}$ を，真の lookahead を未観測文として用いた場合のコンテキスト埋め込みとする．ここでの目標は，コンテキスト埋め込みネットワークが，真の lookahead と同程度に擬似 lookahead を未観測のコンテキスト情報として活用できるようにすることである．したがって，TTS モデルの損失関数に，$\boldsymbol{e}_{\text{pseudo}}$ と $\boldsymbol{e}_{\text{truth}}$ のコサイン類似度を最大化するような損失関数 L_{sim} を追加する．

$$\alpha_{\text{sim}} \cdot L_{\text{sim}} = \alpha_{\text{sim}} \cdot (1 - Sim\,(\boldsymbol{e}_{\text{pseudo}}, \boldsymbol{e}_{\text{truth}})) \quad (4)$$

ここで，パラメータ α_{sim} は L_{sim} の重みであり，$Sim(\cdot)$ はコサイン類似度を表す．このとき，3.2 節の学習過程とは異なり，Tacotron2 の encoder・decoder のモデルパラメータを固定し，コンテキスト埋め込みネットワークのモデルパラメータのみを更新する．これらの一連の処理に基づく fine-tuning を行うことにより，TTS モデルが，言語モデルで生成した擬似 lookahead を考慮した上でコンテキスト情報を捉えることができる．

3.4 提案手法に関する議論

まず，GPT2 で生成した擬似 lookahead が真の lookahead に対してどの程度近いかを議論する．各タイムステップ t について，擬似 lookahead を用いた場合のコンテキスト埋め込み $\boldsymbol{e}_{\text{pseudo}}$ と真の lookahead を用いた場合のコンテキスト埋め込み $\boldsymbol{e}_{\text{truth}}$ とのコサイン類似度を求め，test データ内の平均コサイン類似度を算出した．このコサイン類似度が高い場合，擬似 lookahead は，実際に真の lookahead を観測した場合と同様の効果を合成音声に対してもたらすことが期待できる．さらに，GPT2 のサンプリング手法についても調査する．GPT2 では，確率分布 $p(\boldsymbol{w}_{Nt+1:Nt+L}|\boldsymbol{w}_{1:Nt})$ からサンプリングを行う際に，最も確率値の高い k 単語からランダムにサンプリングする top-k sampling [17] を用いる．したがって，k の値を大きくした場合は，GPT2 は多様な単語候補からサンプリングを行うことになり，$k = 1$ の場合は，最尤基準に基づいて決定論的なサンプリングを行うことになる．

図 3 に分析結果を示す．ここで，学習データ・前処理条件・学習条件には 4.1 節と同一のものを用いた．図 3 中の “top_k ($k = K$)” のラベルは，3.3 節の fine-tuning を用いずに，$k = K$ の top-k sampling を適用したケースを表している．“top_k ($k = 1$, fine-tuned)” は，3.3 節の fine-tuning を適用し，$k = 1$ の top-k sampling を用いたケースを表す．また，“random” は，言語モデルを用いずに，未観測文としてランダムな英単語を与えたケースである．まず，“top_k ($k = K$)” と “random” を比較すると，GPT2 で擬似 lookahead 生成をした場合は，全ての k で “random” よりも高いコサイン類似度を示しており，GPT2 による擬似 lookahead 生成の有効性が確認できる．さらに，k の値が小さくなるごとに，擬似 lookahead によるコンテキスト埋め込みが，真の lookahead によるコンテキスト埋め込みに近づくことが見て取れる．直感的には，k の値を大きくすると，より多様な文章生成が促され，k の値を小さくすると，より客観的にもっともらしい文が生成されることが予想される．この分析結果より，一般的な読み上げコーパスでは，k の値を小さく取る必要があることが示唆される．また，“top_k ($k = 1$, fine-tuned)” の結果より，fine-tuning を行った際のコサイン類似度は，$t = 1, 2$ のケースでは fine-tuning を行わない全てのケースよりも高く，t が大きくなると，fine-tuning を行わないケースより低くなることがある．この fine-tuning 手法では，GPT2 で生成した擬似 lookahead を学習時に考慮するため，t が小さく入力文が十分に観測されていないケースでは，真の lookahead を用いた場合により近いコンテキスト埋め込みが推定できると考えられる．しかし，t が大きくなると，発話文が十分に入力されることで，当該文セグメントの前のコンテキスト情報が増えるため，fine-tuning を用いた場合も fine-tuning を用いない場合と同程度のコサイン類似度に収束すると考えられる．

4. 実験的評価

4.1 実験条件

データセットには LJSpeech [18] を用いた．LJSpeech は，単一の女性英語話者による 13100 文の発話データ (約 24 時間) からなる．発話文の中から 100 発話文・500 発話文を選んでそれぞれ評価セット・テストセットとし，残りを訓練セットとして用いた．サンプリング周波数は 22.05 kHz とした．各々のオーディオデータに短時間フーリエ変換を適用してメルスペクトログラムを抽出する際は，1024 サンプルのフレームサイズ・256 サンプルのホップサイズ・80 チャンネルのメルフィルタバンク・Hanning 窓を用いた．学習時の前処理として 3.2 に示した sliding text window を適用し，窓長を 3・ホップサイズを 1 とした．推論時は，入力セグメント $\boldsymbol{w}_{N(t-1)+1:Nt}$ の単語数 N を 2 に設定した．目標データの当該音声セグメントを forced alignment

表 1 4.2 節に示す各手法の CER・WER・MOS の評価結果.

Methods	CER	WER	MOS
Groundtruth	5.1 %	17.9 %	4.28 ± 0.13
Fullsentence	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

で抽出する際は，Kaldi に基づくツールキット [19] を用いた．評価には，学習済みの GPT2[*1]・WaveGlow[*2]を用いた．GPT2 によるサンプリングを行う際は，全ケースで $k = 1$ の top-k サンプリングを適用した．3.2 節で述べた TTS モデルの一貫学習の際は，バッチサイズを 160 とし，NVIDIA V100 GPU を 4 枚用いて 76000 iteration だけ学習させた．また，3.3 節の fine-tuning を用いる際は，バッチサイズを 32 とし，NVIDIA Geforce GTX 1080Ti GPU1 枚を用いて，コンテキスト埋め込みネットワークのモデルパラメータのみ更新した．この際，$\alpha_{\text{sim}} = 10^{-3}$ とし，4000 iteration だけ学習させた．TTS モデルの一貫学習・fine-tuning のいずれも，Adam [20] を用いて最適化を行い，$\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-6}$ と設定した．TTS モデルの一貫学習の際の学習率を 10^{-3}，fine-tuning の際の学習率を 10^{-4} に設定し，重みを 10^{-6} として L_2 正則化を適用した．

4.2 比較手法

提案手法の有効性を検証するため，複数の比較手法を設定し，客観評価実験・主観評価実験を実施した．比較手法には，(1) ***Groundtruth***: テストデータに含まれる目標発話音声，(2) ***Fullsentence***: 発話文単位の Tacotron2 モデル [4]，(3) ***Independent***: 前後のコンテキストと独立に当該音声セグメントを生成するインクリメンタル TTS 手法 [11]，(4) ***Unicontext***: 過去の既観測文のみをコンテキストとして考慮したインクリメンタル TTS 手法，(5) ***Bicontext***: 本研究の提案手法であり，かつ fine-tuning を適用しない手法，(6) ***Bicontext (fine-tuned)***: 本研究の提案手法であり，かつ fine-tuning を適用した手法，(7) ***Bicontext (truth)***: lookahead-k policy [12] と同様に，真の lookahead を未観測文として考慮した手法，の計 7 手法を設定した．各手法を用いて生成した音声サンプル[*3]は著者の web サイト上で公開されている．

4.3 客観評価実験

インクリメンタル TTS は，発話文単位の TTS と比較して生成に失敗しやすく，発話内容の識別が困難な音声が出力されることがある．したがって，ASR モデルを用いて単語誤り率 (word error rate: WER) と文字誤り率 (character error rate: CER) を算出し，出力音声が人間の発話音声としてどの程度自然で，かつ認識が容易であるかを評価した．ESPnet [21] を使用し，Librispeech [22] で学習された joint-CTC Transformer model [23] を用いて誤り率の算出を行った．表 1 に結果を示す．

まず，***Independent*** が最も低いスコアを示していることが確認できる．***Independent*** は前後のコンテキストを全く考慮しないため，発話文によっては stop flag の予測ができず，非常に間延びした発話音声を出力することがある．そのため，insertion rate の増大に伴って CER・WER ともに大きくなる．結果的に，提案手法の ***Bicontext*** は，***Independent*** よりも顕著に認識が容易で自然な発話音声を生成できていることがわかる．さらに特筆すべき結果として，***Bicontext*** の誤り率は，過去の既観測文のみを考慮する ***Unicontext*** の誤り率よりも小さくなっていることが確認できる．この結果より，擬似 lookahead を未観測のコンテキストとして考慮することが有効に働くことが確認できる．最後に，***Bicontext (fine-tuned)*** と ***Bicontext (truth)*** の結果より，言語モデル誘導型学習による fine-tuning を用いることで，真の lookahead を未観測のコンテキストとして考慮したケースと同程度にまで認識率を改善できることがわかる．

4.4 主観評価実験

合成音声の自然性を評価するため，mean opinion score (MOS) による主観評価実験を実施した．英語を母国語とする 40 人の評価者が Amazon Mechanical Turk [24] を通じて実験に参加し，評価者 1 人つき 35 個の音声サンプルを評価した．この音声サンプルは，7 手法のそれぞれについて，テストセットからランダムに選択した 5 つのサンプルを含んでいる．表 1 に，MOS の平均値と 95%信頼区間を示す．

まず，提案手法の ***Bicontext*** は，当該音声セグメントを独立に生成 [11] する ***Independent*** よりも顕著に高品質であることが確認できる．さらに，***Bicontext*** は ***Unicontext*** よりも有意に高いスコアを示しており，GPT2 による擬似 lookahead がインクリメンタル TTS の自然性を有意に改善することが確認できる．また，提案手法である ***Bicontext*** と ***Bicontext (fine-tuned)*** を比較すると，平均的なスコアとしては ***Bicontext (fine-tuned)*** が上回っており，言語モデル誘導型学習により，擬似 lookahead を未観測のコンテキストとしてより効果的に考慮できることが示唆される．さらに，提案手法は，Ma らの手法 [12] のように，未観測のコンテキストとして真の lookahead を用いる ***Bicontext (truth)*** と同程度のスコアを示している

[*1] https://github.com/graykode/gpt-2-Pytorch
[*2] https://github.com/NVIDIA/waveglow
[*3] https://takaaki-saeki.github.io/itts_lm_demo/

ことがわかる．このことから，提案手法により，未観測文の待機時間を発生させずに，未観測文の入力を待つケースと同程度にまでインクリメンタル TTS の自然性を改善できることがわかる．

5. おわりに

本稿では，大規模言語モデルを用いて擬似 lookahead を生成することで，入力遅延を発生させずに未観測のコンテキストを考慮するインクリメンタル TTS 手法を提案した．本手法は人間の逐次読み上げ機構を模倣するものとして捉えることができ，未観測文の入力を実際に待つ場合と同程度の合成音声品質を達成できることを客観評価実験・主観評価実験により示した．

謝辞 本研究の一部は，JSPS 科研費 17H06101 の助成，及び総務省 SCOPE (受付番号 182103104) の委託を受け実施した．

参考文献

[1] S. Bangalore, V. K. Rangarajan Sridhar, P. Kolan, L. Golipour, and A. Jimenez, "Real-time incremental speech-to-speech translation of dialogs," in *Proc. NAACL*, Montreal, Canada, Jun 2012, pp. 437–445.

[2] K. Sudoh, T. Kano, S. Novitasari, T. Yanagita, S. Sakti, and S. Nakamura, "Simultaneous speech-to-speech translation system with neural incremental ASR, MT, and TTS," *arXiv*, vol. abs/2011.04845, 2020.

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," `https://openai.com/blog/better-language-models/`, 2019.

[4] 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. ICASSP*, Calgary, Canada, Apr. 2018, pp. 4779–4783.

[5] Y. Cong, R. Zhang, and J. Luan, "PPSpeech: Phrase based parallel end-to-end TTS system," *arXiv*, vol. abs/2008.02490, 2020.

[6] K. Tokuda, H. Zen, and A. W. Black, "An HMM-based speech synthesis system applied to English," in *Proc. IEEE WSS*, Santa Monica, U.S.A., Sep. 2002, pp. 227–230.

[7] H. Zen, K. Tokuda, and A. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[8] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, Vancouver, Canada, May 2013, pp. 7962–7966.

[9] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, Ron J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," *arXiv*, vol. abs/1609.03499, 2017.

[10] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proc. AAAI*, Honolulu, U.S.A., July 2019, pp. 6706–6713.

[11] T. Yanagita, S. Sakti, and S. Nakamura, "Neural iTTS: Toward synthesizing speech in real-time with end-to-end neural text-to-speech framework," in *Proc. SSW*, Vienna, Austria, Sep. 2019, pp. 183–188.

[12] M. Ma, B. Zheng, K. Liu, R. Zheng, H. Liu, K. Peng, K. Church, and L. Huang, "Incremental text-to-speech synthesis with prefix-to-prefix framework," in *Proc. EMNLP*, Online, Nov. 2020, pp. 3886–3896.

[13] B. Stephenson, L. Besacier, L. Girin, and T. Hueber, "What the future brings: Investigating the impact of lookahead for incremental neural TTS," in *Proc. Interspeech*, Online, Oct. 2020, pp. 215–219.

[14] D. S. R. Mohan, R. Lenain, L. Foglianti, T. H. Teh, M. Staib, and A. Torresquintero, "Incremental text to speech for neural sequence-to-sequence models using reinforcement learning," in *Proc. Interspeech*, Online, Oct. 2020, pp. 3186–3190.

[15] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," *arXiv*, vol. abs/1811.00002, 2018.

[16] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style Tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *arXiv*, vol. abs/1803.09017, 2018.

[17] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," in *Proc. ACL*, Melbourne, Australia, July 2018, pp. 889–898.

[18] K. Ito and L. Johnson, "The LJ Speech Dataset," `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[19] R. M. Ochshorn and M. Hawkins., "Gentle: A robust yet lenient forced aligner built on Kaldi," `https://lowerquality.com/gentle/`, 2017.

[20] D. Kingma and B. Jimmy, "Adam: A method for stochastic optimization," *arXiv*, vol. abs/1412.6980, 2014.

[21] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," *arXiv*, vol. abs/1804.00015, 2018.

[22] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, South Brisbane, Australia, Apr. 2015, pp. 5206–5210.

[23] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. ICASSP*, New Orleans, U.S.A., Mar. 2017, pp. 4835–4839.

[24] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data?," *Perspectives on Psychological Science*, vol. 6, no. 1, pp. 3–5, 2011.