# Alternate Endings: Improving Prosody for Incremental Neural TTS with Predicted Future Text Input

Brooke Stephenson, Thomas Hueber, Laurent Girin, Laurent Besacier

# Alternate Endings: Improving Prosody for Incremental Neural TTS with Predicted Future Text Input

*Brooke Stephenson*[1,2]*, Thomas Hueber*[1]*, Laurent Girin*[1]*, Laurent Besacier*[2,3]

[1]Université Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France
[2]LIG, UGA, G-INP, CNRS, INRIA, Grenoble, France
[3]NAVER LABS Europe, France

brooke.stephenson@gipsa-lab.grenoble-inp.fr, thomas.hueber@gipsa-lab.grenoble-inp.fr,
laurent.girin@gipsa-lab.grenoble-inp.fr, laurent.besacier@univ-grenoble-alpes.fr

## Abstract

Inferring the prosody of a word in text-to-speech synthesis requires information about its surrounding context. In incremental text-to-speech synthesis, where the synthesizer produces an output before it has access to the complete input, the full context is often unknown which can result in a loss of naturalness. In this paper, we investigate whether the use of predicted future text from a transformer language model can attenuate this loss in a neural TTS system. We compare several test conditions of next future word: (a) unknown (zero-word), (b) language model predicted, (c) randomly predicted and (d) ground-truth. We measure the prosodic features (pitch, energy and duration) and find that predicted text provides significant improvements over a zero-word lookahead, but only slight gains over random-word lookahead. We confirm these results with a perceptive test.

**Index Terms**: Incremental text-to-speech, prosody, neural language models

## 1. Introduction

In incremental text-to-speech synthesis (iTTS), the system starts to output chunks of synthetic audio before the full text input is known [1, 2, 3, 4]. The missing input information often hinders the ability to produce a natural sounding speech sequence, mostly because prosodic features that will be determined by the future context (i.e. the remaining words in the sentence) have not yet been specified. Fortunately, the future input is not completely random; human language is characterized by several lexical and syntactic patterns, which can be statistically learnt and then predicted to a certain extent. Recent advances in language modelling, namely the use of transformer models such as BERT [5] and GPT-2 [6] give us accurate representations of the probability distribution of future words. If this information can be mobilized to fill in the missing data for an iTTS system, it may be possible to retain naturalness while minimizing latency.

Early work in iTTS was conducted in the context of HMM-based models, where linguistic and phonological features extracted from text were used to estimate speech parameters. [7] studied the effects of missing future features by replacing decision tree split criteria with default values and evaluating the degradation in speech quality: while cepstral and aperiodicity features could be estimated fairly accurately with just a local context, prosodic features (f0 and duration) were found to be more dependent on longer range context. Using a similar default value strategy, [2] studied symbolic intonation assignment in the presence/absence of word, phrase and utterance level features. They report that phrase and utterance final words benefit the most from the inclusion of phrase and utterance level features in prosodic assignment determination. [3] explicitly specified unknown features in the context clustering process and found improvements over a default value strategy. [1] tested just-in-time strategies for integrating future chunks in a dialogue system and concluded that incorporating the next phrase once the first word of the current phrase had been processed gave the best latency/quality trade-off. Finally [4] developed an adaptive policy which delayed synthesis when there was high uncertainty regarding POS tags.

Recent research in iTTS has focused on end-to-end neural models. While these models create more natural speech, they are also more difficult to analyze because the relevant features for the task are learnt during training and are subsequently not easily human interpretable. In this new paradigm, the trade-off between speech quality and synthesis latency has been examined by testing the effects of different degrees of lookahead [8, 9], reinforcement learning has been used to automatically learn a wait/synthesize policy [10], the effects of synthesis unit have been studied [11] and the integration of iTTS into a speech-to-speech translation system [12] and into a machine speech chain [13] have been evaluated.

Furthermore, previous studies in conventional (i.e. non incremental) TTS have incorporated language model representations into neural TTS training and found that they could help speed up training time [14, 15]. In the field of simultaneous translation, where future context is also unknown, [16] hallucinated future words to balance the latency/quality trade-off.

In the present work, we propose an iTTS system that incorporates a language model to predict future lookahead.[1] Our approach (described in Figure 1) predicts one word into the future. This limited lookahead was chosen so that the effects of correct and incorrect predictions could be studied. We evaluate our system by contrasting different future word contexts: (a) unknown, (b) language model predicted, (c) randomly predicted (a control group) and (d) ground-truth (see Table 1). Differences are measured at the TTS encoder level and from the generated speech signal through a listening test.

## 2. Method

### 2.1. Definitions

For each token in our corpus, we prepare different sequences which are used as input to the TTS model, FastSpeech 2 [18].

- $\mathbf{x}_{1:n} = x_1, x_2, ..., x_n$ is the sequence of tokens up to $n$. In the

[1]A similar idea was proposed in a contemporary preprint paper [17]. In their work, a context encoder architecture is used.


Figure 1: *Utilizing language model predictions to improve incremental TTS quality while keeping limited latency.*

Table 1: *Examples of input sequences with unknown, ground-truth, predicted and random future context. In each sequence, the word in bold is the word which is synthesized from the sequence.*

| **Input Type** | **Lookahead** | **Input Sequences** |
|---|---|---|
| Ground Truth | Full sentence, $k = N - n$ | **Do you think that you could manage, Tidy?** |
| Unknown (future) | $k = 0$ word | $\mathbf{s}^{\text{GT}}_{1:n+0}$ = **Do**, Do **you**, Do you **think**, ... |
| Ground Truth | $k = 1$ word | $\mathbf{s}^{\text{GT}}_{1:n+1}$ = **Do** you, Do **you** think, Do you **think** that, ... |
| GPT-2 prediction | $k = 1$ word | $\mathbf{s}^{\text{Pred}}_{1:n+1}$ = **Do** they, Do **you** agree, Do you **think** this, ... |
| Random | $k = 1$ word | $\mathbf{s}^{\text{Rand}}_{1:n+1}$ = **Do** dance, Do **you** until, Do you **think** art, ... |

proposed iTTS system, the tokenization policy is to split the sentence on space characters, and then synthesis is triggered when a space character is encountered.

- $k$ is the lookahead parameter (number of future tokens available when synthesizing token $x_n$).
- $\mathbf{s}_{1:n+k} = \{x_1, x_2, ..., x_n, \hat{x}_{n+1}, ..., \hat{x}_{n+k}\} = \{\mathbf{x}_{1:n}, \hat{\mathbf{x}}_{n+1:n+k}\}$ is the sequence used for the synthesis of token $x_n$, where for the ground-truth condition (GT) $\hat{\mathbf{x}}_{n+1:n+k} = \mathbf{x}_{n+1:n+k}$, for the prediction condition (Pred) $\hat{\mathbf{x}}_{n+1:n+k}$ is given by the language model, and for the random condition (Rand) $\hat{\mathbf{x}}_{n+1:n+k}$ is random. The random token generation is described in Section 3.1.
- $\mathbf{s}_{1:n}$ is the input prompt used to generate language model predictions.
- Near the end of the sequence, we replace $n+k$ with $\min(n+k, N)$ where $N$ is the length of the full utterance.

### 2.2. Models

**Language model used for prediction.** We use the GPT-2 language model for our study. This is an auto-regressive model trained to predict the next word given a sequence of past words (causal language modeling task), based on a Transformer architecture. The original GPT-2 [6] is large (1.5B parameters) and since our intended use requires fast predictions, we opted to use a smaller version of GPT-2, called "distilled GPT-2" [19].[2] This model has been trained to produce the same output probability distribution as the original GPT-2 but using fewer layers/parameters.

**TTS model.** For TTS we select a fast and high-quality end-to-end model: FastSpeech 2. The implementation we use [20],[3] trained on the LJ Speech Dataset [21], takes characters as input and converts them to phonemes. Phoneme embeddings are passed through several self-attention layers before the model makes duration, pitch and energy predictions for each phoneme. These feature predictions and the latent phoneme representations are then passed to the decoder (more self-attention layers) which produces a Mel-spectrogram.[4] The Mel-spectrogram is then input into a Parallel WaveGAN vocoder [22] (trained on full sentence inputs) for waveform generation. This model is well suited to iTTS because (1) it is fast which is desirable when the objective is to reduce latency (the speed is achieved by predicting all Mel-spectrogram frames in parallel), and (2) it makes explicit duration predictions for each phoneme, which makes it possible to segment words and only synthesize the word(s) of interest.

### 2.3. Incremental synthesis (iTTS)

We implement an incremental synthesis procedure where each token $x_n$ is synthesized from the input sequence $\mathbf{s}_{1:n+k}$. Mel-spectrogram frames corresponding to individual tokens are identified using the internal duration predictions made by FastSpeech 2. Successive word-level Mel-spectrograms are input into the Parallel WaveGAN vocoder on a word-by-word basis. Resulting waveforms are concatenated together using a 1-ms crossfade to eliminate glitches (synthetic audio samples are available at https://tinyurl.com/ae4nzs).

## 3. Experiments

### 3.1. Corpus and predictions

The English corpus we use for analysis consists of 1,000 sentences from LibriTTS [23]. Sentence length ranges from 5 to 42 words, with a total of 16,965 tokens and 62,556 phonemes.

For each token $x_n$ in the corpus, we sampled five GPT-2 and five random next word predictions ($\hat{x}_{n+1}$). The GPT-2 predictions are constrained to the 30 most likely next words (top-30 sampling strategy). The random words were selected from a list of 1,266 of the most common words in English [24]. Importantly, we force GPT-2 predictions and random predictions to have comparable lengths in term of characters/phonemes because (1) GPT-2 tends to predict shorter words because they are more frequent, (2) in our previous study [8], we found that longer future words have more influence on the current token's internal representation (in a seq-to-seq model) than shorter ones, (3) otherwise, our results may be biased by the fact that the random condition simply has more future context. To control for word length in the random condition, we (1) took the word-length distribution of GPT-2 predictions, (2) randomly

[2] https://huggingface.co/distilgpt2
[3] https://github.com/espnet/espnet
[4] For implementation details, see https://tinyurl.com/s7p38hcr

sampled a word-length category from this distribution (e.g. 2-4 characters), (3) limited our most-common list to only words in this category and (4) randomly sampled a word from this list using a uniform distribution.

GPT-2 uses byte pair encoding (BPE) which breaks words down into subword units to better handle out-of-vocabulary tokens. As such, some of its predictions extend the final prompt word rather than predicting a new token (e.g. previous → previously). To avoid such distortions to our input text, we sample until the first character in the predicted text is a space. This also prevents erroneous punctuation marks from being predicted.

### 3.2. Metrics

**FastSpeech 2 representations.** We aim at evaluating the prosody obtained in the different test conditions: no context ($k = 0$), ground-truth context (GT), predicted context (Pred), random context (Rand). For this aim, we compare the pitch, duration and energy values produced in those conditions with the values produced in the *reference* condition (Ref) where the full context (full sentence input) is used. In the present paper, we concentrate on the case $k = 1$ (one-word lookahead).

As for duration and energy, they are first computed at the phoneme level, using the FastSpeech 2 internal predictions (see Figure 2 for a plot of duration values from an example sentence). A phoneme duration is defined as (the log of) the number of Mel-spectrogram frames of that phoneme. The energy is the squared magnitude of the short-time Fourier transform (STFT), averaged across all frequency bins and across the duration of the phoneme. Then the mean absolute error (MAE) is computed by averaging the absolute value of the difference of duration values obtained in each test condition and in the reference condition across all phonemes of the dataset, and the same for the energy feature. The results are reported in Table 2.

Pitch is evaluated at the sentence level.[5] We first align the Mel-spectrograms obtained in the test and reference conditions with Dynamic Time Warping using the Librosa library [25]. Then we extract the pitch curves from the concatenated audio (see Section 2.3) using Praat/Parselmouth [26, 27] and we compute the MAE in cents between the aligned $f_0$ trajectories:

$$MAE = 1200/T \sum_{t=1}^{T} \left| \log_2 \left( f_0^{Test}(t) / f_0^{Ref}(t) \right) \right|. \quad (1)$$

Then the sentence-level MAEs are averaged across all sentences of the dataset. The results are reported in Table 3.

**Perceptive test.** Finally, we evaluate the global quality using 40 native English speaking evaluators[6] and a MUSHRA test [28]. We selected 20 sentences from our corpus and for each sentence, we presented the listeners with a reference audio clip (generated with the full sentence context) and then asked them to assign a similarity score to five test clips: the hidden reference (identical to the reference and used as the MUSHRA high anchor), $k = 0$ (used as the low anchor), Ground-Truth $k = 1$, GPT-2 prediction $k = 1$ and random prediction $k = 1$. We then compare the distributions of the similarity scores. The responses from four of the participants were removed because these listeners consistently failed to assign a high similarity score to the high anchor. See Figure 3 for results.

[5] We did not evaluate error in the internal FastSpeech 2 pitch predictions because we observed a few extreme prediction values which did not materialize in the resultant audio.

[6] Anonymous participants were recruited using Prolific (www.prolific.co). They were compensated at a rate slightly above the UK minimum wage.

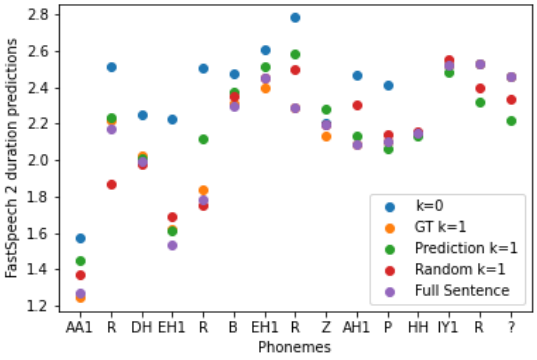


Figure 2: *Duration prediction from FastSpeech2 (in number of Mel-spectrogram frames on a log scale) for each phoneme in the sentence "Are there bears up here?" and for the different tested prediction conditions.*

Table 2: *MAE (and standard deviation across phonemes) between duration (resp. energy) obtained with full context and with limited context. *unit = number of Mel-spectrogram frames on a log scale; **arbitrary unit: signal is digital, normalized and averaged.*

| Input type | # phonemes | Duration* | Energy** |
|---|---|---|---|
| $k = 0$ | 62,556 | $0.262 \pm 0.297$ | $0.301 \pm 0.364$ |
| GT $k = 1$ | 62,556 | $0.077 \pm 0.133$ | $0.176 \pm 0.241$ |
| Pred $k = 1$ | 5×62,556 | $0.135 \pm 0.198$ | $0.247 \pm 0.296$ |
| Rand $k = 1$ | 5×62,556 | $0.147 \pm 0.208$ | $0.260 \pm 0.304$ |
| Correct pred. | 38,274 | $0.086 \pm 0.132$ | $0.187 \pm 0.239$ |
| Incorrect pred. | 274,506 | $0.142 \pm 0.205$ | $0.255 \pm 0.301$ |

Table 3: *MAE between the pitch curves obtained with the full context and with limited context.*

| Input type | # sentences | Pitch MAE (Cents) |
|---|---|---|
| $k = 0$ | 1,000 | $203.56 \pm 45.50$ |
| GT $k = 1$ | 1,000 | $88.57 \pm 26.33$ |
| Pred $k = 1$ | 5×1,000 | $120.03 \pm 29.34$ |
| Rand $k = 1$ | 5×1,000 | $123.03 \pm 30.27$ |

## 4. Discussion

For all metrics, with regards to the mean, we see a clear ranking in the similarity to the full sentence reference: $k = 0$ is farthest away, GT $k = 1$ is the closest and *Pred* and *Rand* are in between, the former being slightly closer to full context than the latter. Statistical tests (t-test for pitch, duration and energy measures and Wilcoxon for the listening test) confirmed that *Pred* and *Rand* do not belong to the same distribution (p-value $< 0.05$) and that *Pred* is better by a small but significant margin.

We notice that duration predictions for $k = 0$ are almost always longer than the other conditions (Figure 2). And as in [1], we observe pitch drops for $k = 0$ words. This is because all words are interpreted as the end of a sentence (as they are the final word in the FastSpeech 2 input, hence sentence final characteristics are predicted by the model). Both the prediction and the random conditions reduce this effect thanks to the additional padding words.

**Correct vs. Incorrect Predictions.** When we separate the

correct from the incorrect GPT-2 next word predictions (see Table 2), we see that the MAE for the incorrect predictions is almost identical to the MAE for the random condition. This suggests that the improved syntactical accuracy gained from the GPT-2 predictions (the POS of the predicted token matches that of the GT next token 43.5% of the time vs. 18.0% for random) does not translate into improved prosodic features.

Since we only see improvement when the exact next word is predicted, it is clear that the minor difference between GPT-2 and random is explainable by the low exact-word prediction rate. We observe that 76% of the GPT-2 sequences have a prediction rate lower than 10%, and 97% have a rate lower than 21% (mean: GPT-2 = 6.8%; random = 0.09%). It is likely that as language models continue to improve [29], we will see greater gains in naturalness from the proposed method (improvements in semantic modelling will narrow the range of word choice, resulting in more frequent exact word predictions). However, these potential advances will have a fairly low ceiling if we consider human prediction abilities as the upper limit: [30] shows that approximately 5% of context words and 20% of function words are highly predictable by humans. Further prediction gains could be achieved if the language model was fine-tuned on the traits of a specific author [31]; this would be an advisable step in the use case of assistive technologies for the speech impaired.

**Context Sensitivity.** Previous studies investigating the impact of lookahead have shown the contrast between $k = 0$ context and different degrees of ground-truth lookahead. The setup of the present study allows us to investigate where the choice of future context modifies the output the most (i.e. where do prosodic features remain stable irrespective of the future context and where do they vary dependent on the future context). To this purpose, we calculated the range of phoneme duration and pitch feature values predicted by the TTS model in all test conditions except $k = 0$. More precisely, from the 12 predicted and ground truth conditions ($5 \times$ Pred, $5 \times$ Rand, GT $k = 1$ and full context), we take the max and min values from this set and calculate the difference. This analysis shows that a large portion of phonemes in the corpus alter only slightly when provided with different next word contexts. The pitch range does not exceed 300 cents for approximately 75% of our samples, which falls below the Just Noticeable Difference (JND) threshold for pitch distance found by [32]. The duration range is limited to a single spectrogram frame (11.75ms) for 40% of phonemes, which, depending on the length of the phoneme, may be imperceptible to the average listener ([33] found a JND of 5%).

We do however see some wide range values in the corpus which explain the large standard deviations in Table 2 and the significant variability of the Pred and Rand scores in the MUSHRA test (Figure 4: the maximum deviation values in a sentence show strong correlation with the mean MUSHRA similarity scores). By examining the corpus, we notice discernible patterns in the locations of large context sensitivity. With respect to pitch, we see large variation when there is a mismatch between predicted and ground truth punctuation at the end of the next word or when there is a reporting verb (e.g. *said*, *exclaimed*) rather than the beginning of a new sentence following a punctuation mark. With respect to duration, the largest variance occurs at the beginning of sentences, at punctuation marks and in function words, especially in the coordinating conjunction *and*.

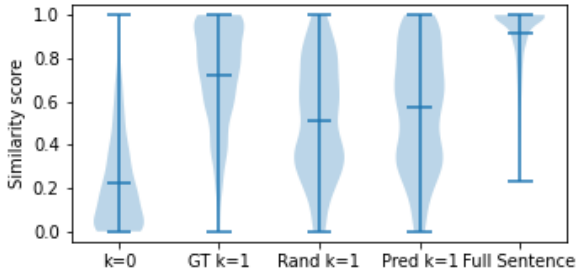


Figure 3: *Violin plots of the distribution of similarity scores between signals generated with full context and signals generated with limited context for the 20 sentences in the MUSHRA test. The middle bars show the mean value.*

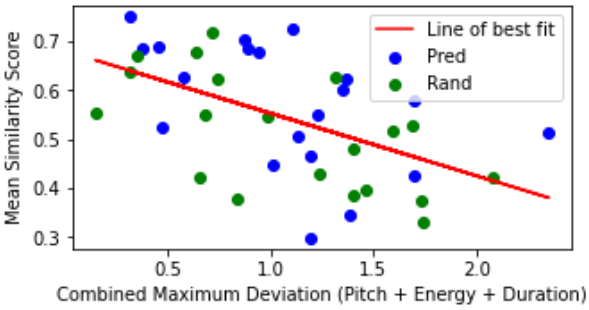


Figure 4: *Each point represents a sentence (synthesized under the Pred or Rand condition) from the MUSHRA test. The x-axis shows the scaled and combined (pitch, energy, duration) maximum deviation values (deviation from the full context value) for the phonemes in the sentence. The y-axis shows the mean similarity score for (Pred,Rand) sentences to their full context counterpart, given by the MUSHRA participants. The Pearson correlation coefficient is equal to* $-0.53$.

## 5. Conclusion and Perspectives

The results from all metrics show that the language model predicted text does improve prosody when compared to the $k = 0$ condition. Slight improvements over the random text condition are also observed. We have seen that language model predictions are often incorrect and context mismatches can occasionally cause major distortions compared to the full context prosody. To improve our model, we could a) implement a wait policy that delays synthesis when a context sensitive word is encountered (similar to [4]) or b) retrain the model on both ground truth and predicted input. With this training regime, the iTTS model would find the optimal solution given contextual ambiguity (similar to [3]); the language model predictions, which frequently differ from the ground truth in terms of relevant contextual features (word choice, next phonemes, number of syllables, POS, and position in the prosodic phrase/utterance) would serve to make the model more robust and perhaps provide more neutral solutions for context sensitive words.

## 6. Acknowledgements

This work was funded by the Multidisciplinary Institute in Artificial Intelligence MIAI@Grenoble-Alpes (ANR-19-P3IA-0003).

# 7. References


[1] T. Baumann and D. Schlangen, "Evaluating prosodic processing for incremental speech synthesis," in *Proc. of Interspeech*, Portland, OR, USA, 2012, pp. 438–441.

[2] T. Baumann, "Partial representations improve the prosody of incremental speech synthesis," in *Proc. of Interspeech*, Singapore, 2014, pp. 2932–2936.

[3] M. Pouget, T. Hueber, G. Bailly, and T. Baumann, "HMM training strategy for incremental speech synthesis," in *Proc. of Interspeech*, Dresden, Germany, 2015, pp. 1201–1205.

[4] M. Pouget, O. Nahorna, T. Hueber, and G. Bailly, "Adaptive latency for part-of-speech tagging in incremental text-to-speech synthesis," in *Proc. of Interspeech*, San Francisco, CA, USA, 2016, pp. 2846–2850.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. of ACL*, Florence, Italy, 2019, pp. 4171–4186.

[6] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, 2019.

[7] T. Baumann, "Decision tree usage for incremental parametric speech synthesis," in *Proc. of ICASSP*, Florence, Italy, 2014, pp. 3819–3823.

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

[9] 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 *Findings of ACL: EMNLP 2020*, 2020, pp. 3886–3896.

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

[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. of SSW*, Vienna, Austria, 2019, pp. 183–188.

[12] 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 preprint arXiv:2011.04845*, 2020.

[13] S. Novitasari, A. Tjandra, T. Yanagita, S. Sakti, and S. Nakamura, "Incremental machine speech chain towards enabling listening while speaking in real-time," *arXiv preprint arXiv:2011.02126*, 2020.

[14] W. Fang, Y.-A. Chung, and J. Glass, "Towards transfer learning for end-to-end speech synthesis from deep pre-trained language models," *arXiv preprint arXiv:1906.07307*, 2019.

[15] Y.-A. Chung, Y. Wang, W.-N. Hsu, Y. Zhang, and R. Skerry-Ryan, "Semi-supervised training for improving data efficiency in end-to-end speech synthesis," in *Proc. of ICASSP*, Brighton, United Kingdom, 2019, pp. 6940–6944.

[16] R. Zheng, M. Ma, B. Zheng, and L. Huang, "Speculative beam search for simultaneous translation," in *Proc. of EMNLP-IJCNLP*, Hong Kong, China, 2019, pp. 1395–1402.

[17] T. Saeki, S. Takamichi, and H. Saruwatari, "Incremental text-to-speech synthesis using pseudo lookahead with large pretrained language model," *arXiv preprint arXiv:2012.12612*, 2020.

[18] 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," *arXiv preprint arXiv:2006.04558*, 2020.

[19] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, and colleagues, "Transformers: State-of-the-art natural language processing," in *Proc. of ACL EMNLP*, Online, 2020, pp. 38–45.

[20] 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 *Proc. of ICASSP*, Barcelona, Spain, 2020, pp. 7654–7658.

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

[22] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proc. of ICASSP*, Barcelona, Spain, 2020, pp. 6199–6203.

[23] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "LibriTTS: A corpus derived from LibriSpeech for text-to-speech," in *Proc. of Interspeech*, Graz, Austria, 2019, pp. 1526–1530.

[24] R. Speer, J. Chin, A. Lin, S. Jewett, and L. Nathan, "Luminosoinsight/wordfreq: v2.2 [Computer software]," Oct. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1443582

[25] Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "Librosa: Audio and music signal analysis in Python," in *Proc. of the Python in Science Conference*, 2015, pp. 18–24.

[26] P. Boersma and D. Weenink, "Praat: doing phonetics by computer [Computer software]," Version 6.0.37, retrieved 3 February 2018 http://www.praat.org/, 2018.

[27] Y. Jadoul, B. Thompson, and B. de Boer, "Introducing Parselmouth: A Python interface to Praat," *Journal of Phonetics*, vol. 71, pp. 1–15, 2018.

[28] ITU-R, "Recommandation BS.1534 and BS.1116: Methods for the subjective assessment of small impairments in audio systems."

[29] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, and colleagues, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[30] S. G. Luke and K. Christianson, "Limits on lexical prediction during reading," *Cognitive Psychology*, vol. 88, pp. 22–60, 2016.

[31] E. Delasalles, S. Lamprier, and L. Denoyer, "Learning Dynamic Author Representations with Temporal Language Models," in *2019 IEEE International Conference on Data Mining (ICDM)*. Beijing, China: IEEE, Nov. 2019, pp. 120–129.

[32] J. 't Hart, "Differential sensitivity to pitch distance, particularly in speech," *The Journal of the Acoustical Society of America*, vol. 69, no. 3, pp. 811–821, 1981.

[33] H. Quené, "On the just noticeable difference for tempo in speech," *Journal of Phonetics*, vol. 35, no. 3, pp. 353–362, 2007.