



# Exploring the Benefits of Tokenization of Discrete Acoustic Units

Avihu Dekel, Raul Fernandez

IBM Research

avihu.dekel@ibm.com, fernanra@us.ibm.com

## Abstract

Tokenization algorithms that merge the units of a base vocabulary into larger, variable-rate units have become standard in natural language processing tasks. This idea, however, has been mostly overlooked when the vocabulary consists of phonemes or Discrete Acoustic Units (DAUs), an audio-based representation that is playing an increasingly important role due to the success of discrete language-modeling techniques. In this paper, we showcase the advantages of tokenization of phonetic units and of DAUs on three prediction tasks: grapheme-to-phoneme, grapheme-to-DAUs, and unsupervised speech generation using DAU language modeling. We demonstrate that tokenization yields significant improvements in terms of performance, as well as training and inference speed, across all three tasks. We also offer theoretical insights to provide some explanation for the superior performance observed.

**Index Terms:** Tokenization, Discrete Acoustic Units, Speech Language Models.

## 1. Introduction

Representations of language, written and spoken, in the form of discrete units provide the foundation for many language-processing tasks. Some historical examples of these inventories have included diphones, phones, and sub-phones (for spoken language) and graphemes and word fragments (for written language). For speech tasks, a classical approach has been to use phonetic representations as a link between text and audio since they can encode prior linguistic knowledge and be perceptually distinctive. More recently, discrete self-supervised representations that we will describe as *Discrete Acoustic Units* (DAUs) have provided an alternative intermediate representation that can exploit learning from very large data resources while dispensing with expert knowledge, and are constructed to retain some of the phonetic attributes that can facilitate intelligibility and reproduce prosodic phenomena. Whether phonetically motivated or self-discovered, these representations play an important role in the pipeline of many text-to-speech (TTS) systems, and their accurate prediction from text remains crucial.

We observe that both phonetic and DAU sequences contain redundancy and predictability as a result of a host of constraints (phonotactical, durational, etc.) and are therefore *compressible*. The utility of compressing long sequences by grouping their constituents into variable-length tokens has been widely acknowledged in fields like Natural Language Processing (NLP), where the naïve approach of operating on raw-character inputs would lead to unnecessarily long sequences, and impose computational constraints in models like Transformers. Similar developments, however, have not yet been as widely adopted when dealing with acoustic units (with a notable recent ex-

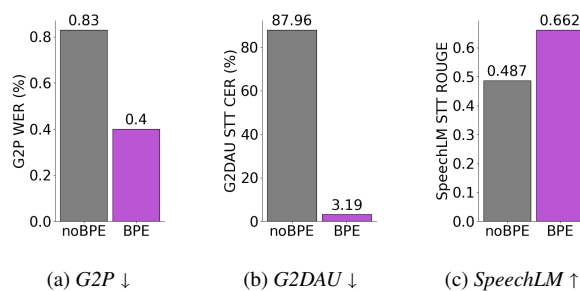


Figure 1: Summarizing the benefits of BPE on DAUs/Phonemes on three tasks. The experimental setup is described in Sec. 4.

ception [1]) despite the more salient need when working with acoustic signals that operate at a much higher bitrate than text, making the processing of a few minutes of audio difficult due to the quadratic complexity of Transformer models.

In this paper, we look closely at the tokenization of phonemes and DAUs, and demonstrate that exploiting this during training is beneficial to learning a task both in terms of training and inference speed as well as the final performance. We document the advantages on three commonly used and important tasks: (a) grapheme-to-phoneme (G2P) conversion, (b) prediction of acoustic units from text (G2DAU), and (c) audio generation using a speech language model (SpeechLM). To probe this, we adopt the Byte Pair Encoding (BPE) algorithm [2, 3], a simple yet effective way to derive a new vocabulary by the iterative grouping of frequent pairs of elements, in order to reduce the length of sequences at the expense of creating a larger vocabulary. Fig. 1 summarizes the main findings of this BPE exploration. To the best of our knowledge, this is the first in-depth performance evaluation of BPE, or other tokenization algorithm, that has been carried out for acoustic units. We provide theoretical insights to shed light on the performance benefits of BPE, including the BPE influence on token imbalance and the connection between sequence length and accuracy in autoregressive models. We hope that these findings will lead to a wider adoption of tokenization algorithms within models that deal with acoustic or phonetic units.

### 1.1. Summary of contributions

Our contributions are three fold:

1. We quantify the compression benefits of applying BPE on discrete audio and phonetic units.
2. We show significant improvements in performance metrics as well as speedups on G2P, G2DAU, and SpeechLM tasks.
3. We show the impact of BPE on mitigating data imbalance and on reducing sequence length in autoregressive models.

## 2. Related Work

**DAUs** are discrete representations of audio signals, usually quantized embeddings from a pre-trained self-supervised speech model (such as HuBERT [4], Wav2Vec2 [5], WavLM [6], or Whisper [7]). Representing continuous-valued high-frequency signals (like speech or audio) with a finite vocabulary of units computed at a much slower rate has led to recent advances in LM techniques when modeling the resulting signals, and to a fruitful field of *audio/speech language modeling* (e.g., [8], AudioLM [9], TWIST [10], and SpeechLM [11]). As DAUs contain important phonetic and suprasegmental information, they have been used as a coarser intermediate representation when predicting the more fine-grained acoustic tokens of a neural codec in discrete TTS systems (like SPEAR [12] and Soundstorm [13]), thus assuming the more classical function of conditional phonetic units, albeit in a purely data-driven way<sup>1</sup>. Such property is also leveraged by other neural (but not discrete) TTS (Tacotron-like) architectures to operate directly on DAU inputs rather than phones [14, 15]. The phonetic and lower bit-rate properties of DAUs have also made them amenable inputs to a lightweight codec that, once augmented with pitch and speaker embeddings, is able to resynthesize and manipulate (e.g., voice-convert) speech [16]. Finally, DAUs have also been used as a proxy for textual representations in unsupervised speech-to-speech translation [17, 18].

**Tokenization** algorithms have been widely adopted in NLP, with a variety of algorithms proposed, including *Word-Piece* [19, 20], sub-word level BPE [21], Unigram [22], and Sentence Piece [23], with BPE arguably being the most common tokenization algorithm due to its simplicity. BPE-based tokenization applied to DAUs was only very recently explored for speech synthesis [1], though that work does not look to isolate the contribution of tokenization. Tokenized units have also been exploited within speech recognition systems, both textually derived (e.g., via BPE as in [24]) and acoustically derived (e.g., the ADSM model of [25]). The successive merging of DAUs within BPE leads to a *variable-rate* inventory, an idea closely related to work on event-driven audio representations [26, 27].

Work in end-to-end TTS has demonstrated the ability of these models to work directly from textual inputs, bypassing explicit intermediate representations with character-to-acoustic models [28]. In practice, however, the superior robustness of phonetic inputs has been documented [29], and many modern state-of-the-art systems continue to rely on phonetic inputs and a separate **G2P** module [30, 31, 32], a fact that continues to fuel development of modern G2P models like T5-G2P [33], Soundchoice [34], ByT5 [35], and LLM2Speech [36]. None of these cited works, however, exploit the advantages of tokenization for G2P prediction that we propose and demonstrate here.

## 3. Methodology

### 3.1. Base Unit Construction

**DAUs:** We extract DAUs following [17] using a pre-trained mHuBERT model, extracting the embeddings from the 11th layer and quantizing them into  $K = 1000$  clusters using pre-trained K-Means centroids<sup>2</sup>. This results in a discrete sequence

<sup>1</sup>While the term *Semantic Token* has been proposed [9], we adopt the term DAU to be more neutral about the nature of these representations, and be consistent with the unit-vs-token nomenclature of Sec. 3.2

<sup>2</sup>Pretrained quantizer and vocoder are available at: [https://github.com/facebookresearch/fairseq/blob/main/examples/speech\\_to\\_speech/docs/](https://github.com/facebookresearch/fairseq/blob/main/examples/speech_to_speech/docs/)

with a frequency of 50 Hz. **Phones:** For G2P experiments, we generate phonetic sequences from text using a proprietary rules-based phonetizer from the linguistic analysis front-end of a TTS system. This module uses an inventory of 45 phones (17 vowels and 28 consonants) with 3 levels of lexical stress per syllable. We extract unique combinations of vowel phones and lexical stress, and merge these with the set of consonant symbols, to arrive at a final phonetic vocabulary containing 81 units (including pause and word separator). Additionally, we include 3 special tokens (PAD, BOS, EOS) for DAUs and phones, to perform autoregressive modeling. **Datasets:** In G2P experiments, we make use of a random subset of the *Common Crawl* [37] dataset, consisting of 3M/6K train/validation paragraphs respectively. Each paragraph was truncated (on sentence ending) to contain at most 200 words. In DAU experiments, we use the English subset of the *multi-lingual LibriSpeech* corpus [38], which contains 10M/3.8K train/validation transcribed utterances of length 10–20 seconds. For the SpeechLM experiments, we included additional training-only data from *People’s Speech* [39], *VoxPopuli* [40], and *Common Voice* [41]<sup>3</sup>.

### 3.2. Byte Pair Encoding

In terms of nomenclature, henceforth a *unit* is an entry in the original inventory whereas a *token* is an entry in the inventory derived via a *tokenization* algorithm that produces non-uniform groupings of the original unit set. To derive the token vocabularies, we make use of the standard BPE algorithm. We start with an original discrete vocabulary  $\mathcal{X}$ , where each sample is a sequence of elements from the vocabulary  $x = (x_1, \dots, x_n)$  s.t.  $x_i \in \mathcal{X}$ . After applying the algorithm described in Alg. 1, we obtain a new vocabulary  $\mathcal{Z}$  where  $|\mathcal{Z}| > |\mathcal{X}|$  and  $x$  can now be represented by  $z = (z_1, \dots, z_k)$  where  $k < n$ . Denoting the model by  $\Theta$  and the context (e.g. text) by  $W$ , we get two equivalent formulations of the learning task using a neural autoregressive model, modeling the original sequence (Eqn. 1) or the BPE-derived sequence (Eqn. 2):

$$p(x_1, \dots, x_n | \Theta, W) = \prod_{i=1}^n p(x_i | \Theta, W, x_1, \dots, x_{i-1}), \quad (1)$$

$$p(z_1, \dots, z_k | \Theta, W) = \prod_{i=1}^k p(z_i | \Theta, W, z_1, \dots, z_{i-1}). \quad (2)$$

---

**Algorithm 1** BPE (Byte Pair Encoding)

---

- 1: **Input:** Sequences, Raw Vocab  $\mathcal{X}$ , Target Size  $k$
  - 2: **Output:** BPE Vocabulary  $\mathcal{Z}$
  - 3:  $\mathcal{Z} = \mathcal{X}$
  - 4: **while**  $|\mathcal{Z}| < k$  **do**
  - 5:     Find the most frequent pair of adjacent units  $(a, b)$
  - 6:     Merge  $(a, b)$  to form a new symbol  $ab$
  - 7:     Add  $ab$  to  $\mathcal{Z}$
  - 8:     Replace all occurrences of  $(a, b)$  with  $ab$
  - 9: **end while**
- 

Why might the formulation of Eqn. 2 be better given the equivalency of the learning tasks? As has already been pointed out by other authors [42], by iteratively merging the most frequent pairs into new tokens, BPE balances the tokens’ distri-

textless\_s2st\_real\_data.md

<sup>3</sup>We thank Ankit Gupta for preparing these datasets.

bution, and it is known that skewed data distributions pose an obstacle to neural models trained with cross-entropy loss.

To quantify this effect, we propose the use of normalized entropy, a balance metric that is invariant to vocabulary size. Given a vocabulary  $\mathcal{X}$  with a probability distribution over its elements  $D(x) : x \in \mathcal{X}$ , and the distribution’s entropy given as  $H(D) = -\sum_{x \in \mathcal{X}} D(x) \log_2(D(x))$ , we define the *normalized entropy* as

$$0 \leq N(D) = \frac{H(D)}{\log_2(|\mathcal{X}|)} \leq 1, \quad (3)$$

where a value of 1 corresponds to a perfectly balanced distribution (i.e., a uniform multinomial). Table 1 illustrates the value of this metric before and after applying BPE to the original phonetic and DAU vocabularies, showing the significant change in balance introduced by BPE.

Table 1: BPE impact on balancing

Domain	BPE	Vocab Size	Balance metric $N(D)$
Phonetic	✗	84	0.797
	✓	256	<b>0.919</b>
DAU	✗	1003	0.876
	✓	2048	<b>0.944</b>

### 3.3. Sequence length in autoregressive models

We can also obtain insights into the performance of BPE-tokenized models by noting the following. An autoregressive model’s error rate accumulates as the sequence gets longer. BPE manages to alleviate this by reducing the sequence length, but it does so while increasing the vocabulary size, effectively making the classification of each individual token harder. The adoption of BPE, therefore, introduces a trade-off between sequence length and token-level accuracy. Consider that in the original vocabulary, a sequence has length  $n_1$  with a token error rate of  $\epsilon_1$ , and, after tokenization, length  $n_2$  with a *token error rate* of  $\epsilon_2$ , and let’s consider the “edge case” of having the model classify every token correctly. Assuming that the average error rate for every token is the same and independently distributed (which is unrealistic), the probability of such, in the  $i^{\text{th}}$  scenario, would be  $P(\text{correct})_i = (1 - \epsilon_i)^{n_i}$ . We illustrate this difference with some actual values from a G2P experiment where the average original sequence length is  $n_1 = 872$  and, after tokenizing with a vocabulary of 2048,  $n_2 = 300$ . The corresponding empirical average errors are found to be  $\epsilon_1 = 0.097\%$  and  $\epsilon_2 = 0.14\%$ , respectively, which leads to  $P(\text{correct})_1 = 42.94\%$  and  $P(\text{correct})_2 = 65.61\%$ . With the tokenized vocabulary, the probability of this edge case is substantially higher.

## 4. Experiments

In the following experiments, we apply BPE with varying vocabulary sizes to both DAU and phonemes, and compare the performance with respect to the original discrete vocabularies by training Transformer models on the following tasks:<sup>4</sup>

1. G2P: Grapheme to Phoneme Prediction (Sec. 4.3)
2. G2DAU: Grapheme to Discrete-Audio-Units (Sec. 4.4)
3. SpeechLMs using Discrete-Audio-Units (Sec. 4.5)

<sup>4</sup>With phonemes, we apply sub-word level tokenization: we do not merge the phonemes of different words.

## 4.1. Evaluation Metrics

### 4.1.1. BPE Evaluation

To evaluate BPE compression, we make use of the following metrics. First, the *reduction in sequence length*, and the relative increase in the number of bits needed to represent the vocabulary, are given by:

$$\text{Reduction} = \frac{\hat{n}}{\hat{k}}, \quad \text{BitIncrease} = \frac{\log_2(|\mathcal{Z}|)}{\log_2(|\mathcal{X}|)}, \quad (4)$$

where  $\hat{n}, \hat{k}$  denote the average length of the original and BPE sequence, respectively. Using those, we define the *compression* achieved by BPE as:

$$\text{Compression} = \frac{\text{Reduction}}{\text{BitIncrease}} = \frac{\hat{n} \log_2(|\mathcal{X}|)}{\hat{k} \log_2(|\mathcal{Z}|)} \quad (5)$$

### 4.1.2. Task Evaluation

For **G2P** we follow standard practice and report Word Error rate (WER)<sup>5</sup>. For the **G2DAU** task, as there are various possible options for a correct DAU translation, we follow the work in SPEAR TTS [12] and calculate the Character Error Rate (CER) obtained with an external Speech-to-Text (STT) system. Specifically, we synthesize the DAU tokens using the pre-trained vocoder described in [17], and apply STT using Whisper-large-v3 to translate the audio back to text, and calculate and report CER between the original text and the STT output.

Finally, for the **SpeechLM** task, we evaluate the generated audio against a selected reference (details in Sec. 4.5) by transcribing it with STT (as above) and computing CER plus two other established metrics from NLP for text comparison: BLEU (for machine translation) and ROUGE (for summarization). We additionally evaluate the quality of the generation by scoring the STT transcripts with the Mixtral8x7B LM [43]. Given a prompt and two continuations, the LM is asked to select which continuation is better, given the following evaluation criteria: The continuation should (a) be not too short (at least a sentence long), (b) not contain repetitions, (c) be a sensible continuation, and (d) be creative. Each comparison was done twice, replacing the transcripts’ ordering. To allow for a qualitative impression of quality and prosody, we provide a page with samples synthesized using the pretrained vocoder<sup>6</sup>.

Finally, we measure speedup gains by reporting the relative increase in the number of batches per second, compared to the baseline model that does not apply BPE. We ensure all computing resources are identical within a set of experiments. Results are reported for the training set, though similar speedups are obtained for inference.

## 4.2. Model Training

We use the *T5-small* Encoder-Decoder architecture (75M parameters) for the G2P experiments, and *T5-base* (280M parameters) for the G2DAU experiments [44]. All models are optimized using AdamW [45] with a batch size of 32, and weight decay of 0.1. The learning rate is linearly increased to  $1e-4$  over 10k warm-up steps, and annealed using a cosine schedule over 400k iterations. For G2P we train with two V100 GPUs, and for G2DAU with two A100 GPUs. All weights are initialized using Xavier initialization, and we use greedy autoregressive decoding for inference. For SpeechLMs, we train

<sup>5</sup>All metrics are computed in the original vocabulary.

<sup>6</sup>Sample page is available here: <https://ibm.biz/BdmLCb>

a decoder-only model based on the LLaMA [46] architecture, with 24 layers of dimensionality 1024, 16 heads per layer, and feed-forward network of size 4096 (400M parameters). Each model is trained with four A100 GPUs with a batch of 64 samples, for 1M iterations. During inference, we sample the next token with a temperature of 1, sampling from the top 20 tokens, using beam search with 4 beams, and a repetition penalty of 1.2.

Table 2: Results of applying BPE to G2P. (First row indicates the original vocabulary.)

Vocab	↑Reduction	↑Compression	↓WER %	↑Speed
84	1	1	0.83	1x
256	1.69	1.35	0.69	1.27x
512	2.03	1.44	0.58	1.48x
1024	2.43	1.55	<b>0.40</b>	1.69x
2048	<b>2.90</b>	<b>1.69</b>	0.46	<b>1.75x</b>

#### 4.3. Task 1: Grapheme to Phonemes

Results in Table 2 show that BPE exploits redundancy in the raw phonetic representation (row 1) in order to compress (rows 2-5). Training and inference time are shorter with BPE tokenization, and accuracy is superior.

Table 3: Results of applying BPE to G2DAU. (First row indicates the original vocabulary). STT CER is also reported for the raw audio (Orig) and the vocoder reconstruction of the ground truth DAUs (Recon).

Vocab	↑Reduction	↑Compression	↓CER %	↑Speed
1003	1	1	87.96	1x
2048	1.89	1.71	9.94	1.95x
4096	2.39	1.98	5.12	2.28x
8192	2.81	2.15	3.32	<b>2.55x</b>
16384	<b>3.20</b>	<b>2.27</b>	<b>3.19</b>	2.35x
Orig	-	-	2.04	-
Recon	-	-	7.41	-

#### 4.4. Task 2 : Graphemes to Discrete Acoustic Units

Results in Table 3 show significant benefits on the CER when applying BPE, with the reduction in sequence length greatly influencing the training and inference speed. One possible explanation for why the advantages of tokenization are more apparent here than for the G2P task lies in the structure of the DAUs: The scale at which they are extracted leads to sequences with frequent repetitions (e.g., *aaabbbbccccdd*). This fact can be exploited during teacher-forcing training by a simple heuristic that copies the prediction of the previous tokens, reducing the next-token-prediction loss. This, however, creates a mismatch with respect to the true-inference condition. BPE reduces repetition, thereby mitigating this exposure bias, and bringing training and inference closer to each other. Note that the model using the largest vocabulary size (16384) is *not* the fastest. At some point, the sequence-reduction rate no longer compensates for the larger matrix multiplication in prediction. Very large vocabularies also increase memory consumption, due to the logits matrix of size  $batch\_size \times seq\_len \times vocab\_size$ . Note also the high CER in the non-BPE case. To allow for fair comparisons, we ensured all models shared the same training conditions, and,

after 400k iterations, this model had not yet reached the high-enough accuracy that AR models require for stable next-token prediction, leading to nonsense sequences and the CER rate reported. This was a stable finding verified across various experiments.

Table 4: Speech-generation metrics without (row 1) and with (row 2) BPE tokenization.

Vocab	↓CER %	↑BLEU	↑ROUGE	↑Speed
1003	67.04	0.134	0.487	1x
16384	<b>63.02</b>	<b>0.172</b>	<b>0.662</b>	<b>1.63x</b>

#### 4.5. Task 3: SpeechLMs

In this task we explore the generation of audio without any textual supervision by a speech LM that takes in an audio prompt and *continues* the audio. To do this, we segment the first 4 seconds of an audio file to act as the prompt, and withhold the rest of it as the *Ground-Truth Continuation* (GTC). We then evaluate the hypothesized completions of the input prompt by applying the same STT protocol described in section 4.1.2 and computing CER, BLEU and ROUGE against the GTC reference (Table 4). It is important to note that there is not a unique way to complete an initial prompt, and that the GTC is but one valid outcome. However, given the difficulty of evaluating reference-free speech generation, we can take closeness to the GTC to be, on average, one measure of the fitness of the hypotheses. The results in Table 4 suggest that the BPE SpeechLM results in better performance than the non-BPE SpeechLM.

Additionally, we evaluate the quality of the continuations using an LLM as a judge [47]. Given a textual prompt and two continuations, the LM judges which continuation is better (see Sec.4.1.2). Table 5 shows the BPE variant is preferable over the non-BPE, and that both still lag behind the GTC.

Table 5: LLM-as-a-Judge evaluation results.

A	B	Prefer A %	Prefer B %
BPE	non-BPE	<b>69.52</b>	30.48
GTC	BPE	<b>77.34</b>	22.66
GTC	non-BPE	<b>93.89</b>	6.11

## 5. Conclusions

In this paper we have investigated the effect of tokenizing inventories of phonetic and discrete audio representations via the BPE algorithm in various tasks that deploy them: their respective prediction from graphemes, and their use within speech-generation models. By quantifying the trade-off between reducing sequence length and increasing vocabulary size for these inventories, and by demonstrating that exploiting BPE consistently outweighs the choice of not including it, both in terms of efficiency and performance, it is our recommendation and hope that practitioners in the field will adopt this or similar tokenization approaches going forward. Future extensions of this work include investigating the effect of using predicted BPE tokens directly into a full TTS system (without unpacking them into their constituent units), and reconciling the variable-rate nature of these tokens with the constant-frame outputs typically generated by TTS architectures.

## 6. References

- [1] M. Łajszczak *et al.*, “BASE TTS: Lessons from building a billion-parameter text-to-speech model on 100k hours of data,” 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2402.08093>
- [2] P. Gage, “A new algorithm for data compression,” *The C User Journal*, vol. 18, 1994.
- [3] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” in *Proc. ACL*, Berlin, Germany, 2016, pp. 1715–1725. [Online]. Available: <https://aclanthology.org/P16-1162>
- [4] W.-N. Hsu *et al.*, “HuBERT: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Trans. on Audio, Speech and Language Processing*, pp. 3451–3460, 2021.
- [5] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *Proc. NeurIPS*, vol. 33, 2020, pp. 12 449–12 460.
- [6] S. Chen *et al.*, “WavLM: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE J. of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.
- [7] A. Radford *et al.*, “Robust speech recognition via large-scale weak supervision,” in *Proc. ICML*, vol. 202, 23–29 Jul 2023, pp. 28 492–28 518.
- [8] K. Lakhotia *et al.*, “On generative spoken language modeling from raw audio,” *Trans. ACL*, vol. 9, pp. 1336–1354, 2021. [Online]. Available: <https://aclanthology.org/2021.tacl-1.79>
- [9] Z. Borsos *et al.*, “AudioLM: a language modeling approach to audio generation,” 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2209.03143>
- [10] M. Hassid *et al.*, “Textually pretrained speech language models,” in *Proc. NeurIPS*, 2023. [Online]. Available: <https://openreview.net/forum?id=UIHueVjAKr>
- [11] Z. Zhang *et al.*, “SpeechLM: Enhanced speech pre-training with unpaired textual data,” 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2209.15329>
- [12] E. Kharitonov *et al.*, “Speak, read and prompt: High-fidelity text-to-speech with minimal supervision,” 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2302.03540>
- [13] Z. Borsos *et al.*, “SoundStorm: Efficient parallel audio generation,” 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2305.09636>
- [14] A. Garg, J. Kim, S. Khyalia, C. Kim, and D. Gowda, “Data-driven grapheme-to-phoneme representations for a lexicon-free text-to-speech,” in *Proc. ICASSP (to appear)*, 2024.
- [15] C. Liu, Z.-H. Ling, and L.-H. Chen, “Pronunciation dictionary-free multilingual speech synthesis using learned phonetic representations,” *IEEE/ACM Trans. on Audio, Speech, and Lang. Proc.*, vol. 31, pp. 3706–3716, 2023.
- [16] A. Polyak *et al.*, “Speech resynthesis from discrete disentangled self-supervised representations,” in *Proc. Interspeech*, 2021, pp. 3615–3619.
- [17] A. Lee *et al.*, “Textless speech-to-speech translation on real data,” in *Proc. ACL-HLT*, Seattle, United States, July 2022, pp. 860–872.
- [18] —, “Direct speech-to-speech translation with discrete units,” in *Proc. ACL*, Dublin, Ireland, 2022, pp. 3327–3339.
- [19] M. Schuster and K. Nakajima, “Japanese and Korean voice search,” in *Proc. ICASSP*, 2012, pp. 5149–5152.
- [20] Y. Wu *et al.*, “Google’s neural machine translation system: Bridging the gap between human and machine translation,” 2016. [Online]. Available: <https://doi.org/10.48550/arXiv.1609.08144>
- [21] A. Vaswani *et al.*, “Attention is all you need,” in *Proc. NeurIPS*, vol. 30, 2017.
- [22] T. Kudo, “Subword regularization: Improving neural network translation models with multiple subword candidates,” in *Proc. ACL*, 2018, pp. 66–75.
- [23] T. Kudo and J. Richardson, “SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” in *Proc. EMNLP*, Nov 2018, pp. 66–71.
- [24] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, “Improved training of end-to-end attention models for speech recognition,” *Proc. Interspeech*, pp. 7–11, 2018.
- [25] W. Zhou, M. ZeinEdein, Z. Zheng, R. Schlüter, and H. Ney, “Acoustic data-driven subword modeling for end-to-end speech recognition,” in *Proc. Interspeech*, 2021, pp. 2886–2890.
- [26] S. Dieleman, C. Nash, J. Engel, and K. Simonyan, “Variable-rate discrete representation learning,” 2021. [Online]. Available: <https://doi.org/10.48550/arXiv.2103.06089>
- [27] M. Lisboa and G. Bellec, “Spiking music: Audio compression with event based auto-encoders,” 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2402.01571>
- [28] W. Ping *et al.*, “Deep Voice 3: 2000-speaker neural Text-to-Speech,” *Proc. ICLR*, vol. abs/1710.07654, 2017.
- [29] J. Taylor and K. Richmond, “Analysis of pronunciation learning in end-to-end speech synthesis,” in *Proc. Interspeech*, 2019, pp. 2070–2074.
- [30] J. Shen *et al.*, “Natural TTS synthesis by conditioning Wavenet on MEL spectrogram predictions,” in *Proc. ICASSP*, 2018, pp. 4779–4783.
- [31] C. Wang *et al.*, “Neural codec language models are zero-shot text to speech synthesizers,” 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2301.02111>
- [32] K. Shen *et al.*, “NaturalSpeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers,” in *Proc. ICLR*, 2024.
- [33] M. Řezáčková, J. Švec, and D. Tihelka, “T5G2P: Using Text-to-Text Transfer Transformer for Grapheme-to-Phoneme Conversion,” in *Proc. Interspeech*, 2021, pp. 6–10.
- [34] A. Ploujnikov and M. Ravanelli, “SoundChoice: Grapheme-to-Phoneme models with semantic disambiguation,” in *Proc. Interspeech*, 2022, pp. 486–490.
- [35] J. Zhu, C. Zhang, and D. Jurgens, “ByT5 model for massively multilingual grapheme-to-phoneme conversion,” in *Proc. Interspeech*, 2022, pp. 446–450.
- [36] A. Dekel *et al.*, “Speak while you think: Streaming speech synthesis during text generation,” in *Proc. ICASSP (to appear)*, 2024.
- [37] “Common Crawl,” <https://commoncrawl.org/overview>.
- [38] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, “MLS: A large-scale multilingual dataset for speech research,” in *Proc. Interspeech*, 2020, pp. 2757–2761.
- [39] D. Galvez *et al.*, “The People’s Speech: A large-scale diverse English speech recognition dataset for commercial usage,” in *NeurIPS Datasets and Benchmarks*, 2021.
- [40] C. Wang *et al.*, “VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation,” in *Proc. ACL*, Aug. 2021, pp. 993–1003.
- [41] R. Ardila *et al.*, “Common Voice: A massively-multilingual speech corpus,” in *Proc. LREC*, May 2020, pp. 4218–4222.
- [42] T. Gowda and J. May, “Finding the optimal vocabulary size for neural machine translation,” in *Proc. EMNLP 2020*. ACL, Nov. 2020, pp. 3955–3964.
- [43] A. Q. Jiang *et al.*, “Mixtral of experts,” *arXiv preprint arXiv:2401.04088*, 2024.
- [44] C. Raffel *et al.*, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *J. Mach. Learn. Res.*, vol. 21, pp. 1–67, 2020.
- [45] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *Proc. ICLR*, 2019.
- [46] H. Touvron *et al.*, “LLaMA: Open and efficient foundation language models,” 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2302.13971>
- [47] L. Zheng *et al.*, “Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.