



A Language Modeling Approach to Diacritic-Free Hebrew TTS

Amit Roth, Arnon Turetzky, Yossi Adi

School of Computer Science and Engineering
The Hebrew University of Jerusalem, Israel

amit.roth@mail.huji.ac.il

Abstract

We tackle the task of text-to-speech (TTS) in Hebrew. Traditional Hebrew contains *Diacritics*, which dictate the way individuals should pronounce given words, however, modern Hebrew rarely uses them. The lack of diacritics in modern Hebrew results in readers expected to conclude the correct pronunciation and understand which phonemes to use based on the context. This imposes a fundamental challenge on TTS systems to accurately map between text-to-speech. In this work, we propose to adopt a language modeling Diacritics-Free approach, for the task of Hebrew TTS. The model operates on discrete speech representations and is conditioned on a word-piece tokenizer. We optimize the proposed method using in-the-wild weakly supervised data and compare it to several diacritic-based TTS systems. Results suggest the proposed method is superior to the evaluated baselines considering both content preservation and naturalness of the generated speech. Samples can be found under the following link: pages.cs.huji.ac.il/adiyoss-lab/HebTTS/

Index Terms: Text-to-Speech, Diacritic, Hebrew speech

1. Introduction

Hebrew, a low-resource language spoken by 9 million people worldwide [1], presents unique challenges that constrain research and product development in speech technology. Specifically, Hebrew is a morphologically rich language, with the common use of prefixes and suffixes to modify words' meanings and to add prepositions. On top of that, Hebrew uses Diacritics ('Niqqud') to create a one-to-one mapping between text and phonemes. 'Niqqud' is a system of Diacritical signs used to represent vowels or distinguish between alternative pronunciations of letters of the Hebrew alphabet.

In practice, modern Hebrew text rarely contains Diacritics, one may find Diacriticized text in specialized texts such as dictionaries, poetry, or children's books. Hence, readers are expected to conclude the correct pronunciation and understand which phonemes to use, based on familiarity with the language itself. This makes it challenging for text-to-speech (TTS) systems to accurately learn the connection between text and speech. For example the words: *מתנה* and *מתנה*, contain the same characters, but have completely different meaning and pronunciation, the first one means a 'gift', while the second one means 'conditioning'. As mentioned before, in modern Hebrew writing, one will probably not encounter Diacritics and the above word will appear as follows, *מתנה*. As a result, the reader should infer the right pronunciation by context only. Moreover, when considering spoken language modeling systems in automated pipelines, current Automatic Speech Recognition (ASR) systems, such as Whisper [2] and Massively

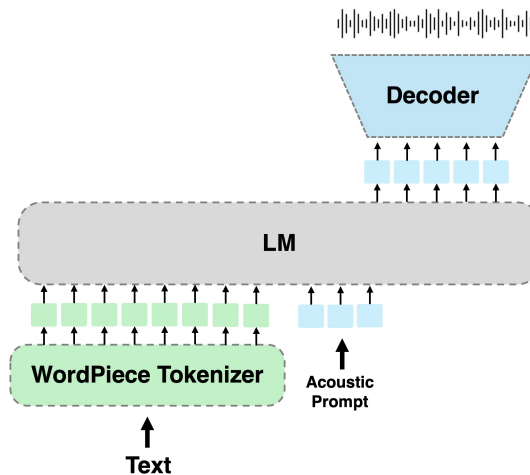


Figure 1: A high-level overview of the proposed method. The text is first being tokenized using a word-piece tokenizer. Then an audio language model predicts a discrete sequence of audio tokens which later on will be decoded into raw waveform.

Multilingual Speech (MMS) [3] do not output diacritics in their transcripts, hence TTS systems should either predict it or use a Diacritic-free synthesis system.

Another major issue that holds progress in developing AI-based Hebrew TTS systems is the lack of datasets. As Hebrew is considered a low-resource language, public spoken benchmarks hardly exist. Previous efforts in constructing datasets in Hebrew were often relatively small [4, 5, 6, 7]. The authors in [4] established the Corpus of Spoken Israeli Hebrew (CoSIH) with the goal of compiling a large database of recordings of spoken Israeli Hebrew in order to facilitate and enhance research in the field. Next, the authors in [5] released The Map Task Corpus (MaTaCOP) of Hebrew dialogues. The authors in [6] collected naturally occurring speech and interaction in Modern Hebrew via telephone conversations during the years 2020–2021 and released the HUJI Corpus of Spoken Hebrew (HUJICorpus). More recently, the authors in [7] released SASPEECH, a high-quality single-speaker Hebrew dataset to enhance Hebrew speech synthesis research. Although all of these prior work are important and valuable, the provided benchmarks are relatively small. CoSIH contains ~ 12.3 hours of speech, the MaTaCOP corpus contains ~ 5.3 hours, the HUJI Corpus has ~ 3.8 , and SASPEECH which is the largest one contains ~ 30 hours of speech. For comparison a modern, contextualized TTS system in English is trained over $\sim 60k$ hours [8]. Recently, the authors of [9] released a dataset denoted as *ivrit.ai*. The authors

released ~ 3300 hours of weakly-supervised speech from local podcasts and provided the first large-scale dataset in Hebrew. We leverage this dataset in our work while augmenting it with additional ~ 1700 hours of in-the-wild recordings using weak supervision.

Previous attempts were made to construct a TTS system in Hebrew. The Authors of [3], proposed the MMS system. In their study, they develop speech technologies (ASR, TTS, Language ID) in more than 1,000 languages. Their TTS system is based on representation obtained from a pre-trained multi-lingual self-supervised model. Although providing impressive results, their Hebrew TTS system is based on predicting diacritics of the input text. More recently, the authors of [7] introduced the Overflow [10] model for Hebrew, together with the SPASEECH benchmark. The Overflow model is comprised of neural HMM together with normalizing flows. On top of the Overflow model, the authors in [7] suggested using the HiFi-GAN neural vocoder [11] to estimate the phase. Similarly to MMS, the system proposed by [7] is based on predicting diacritics of the input text, hence is sub-optimal and often produces wrong and unnatural content in the generated speech. Moreover, such dependency makes it difficult for these models to scale to large datasets as they both require predicting diacritics on top of automatically transcribed text. Unlike these methods, the proposed LM approach operates in a Diacritic-free manner, not propagating mistakes from the diacritic prediction models, and better leveraging the context of the input signal.

Recent studies in speech and audio representation learning proposed learning discrete representation of the input signal [12, 13]. Such representation can be later used for plenty of speech and audio synthesis tasks [14, 15, 16, 17]. Specifically, the authors of [8, 18, 19] proposed optimizing an LM on top of such discrete speech representation, conditioned on a phonemic representation of the input text for the task of TTS. Following such an approach was found to produce high-quality and natural speech, with the ability to rapidly adapt to new speakers via acoustic prompting. As this approach is contextualized by nature it may serve as the ideal candidate for a Diacritic-free Hebrew TTS system.

In this work, we study and propose a *Language Modeling* approach which operates over discrete representations of the speech signal to construct a Hebrew TTS system. We optimize an acoustic LM over a weakly supervised large-scale dataset containing in-the-wild recordings. We empirically demonstrate that following the LM approach makes the usage of diacritics in Hebrew redundant, hence yielding a diacritic-free approach. We study several text representation methods and found that using word-piece tokenization produces the best results overall. Results suggest the proposed method is superior to the evaluated baselines considering both content preservation and generation quality. Code, dataset, and models are publicly available under the following link: <https://pages.cs.huji.ac.il/adiyoss-lab/HebTTS/>.

2. Method

Given a dataset $D = \{x_i, y_i\}$ where y_i is an audio sample and x_i is its corresponding transcription. We encode the audio into a sequence of discrete units. We follow the approach proposed by [12] and encode the audio using Residual Vector Quantization (RVQ). Formally, $E(y) = C^{T \times N_{cb}}$ where C represents the acoustic code matrix, where N_{cb} is the number of codebooks and T is the utterance length.

A common paradigm in the TTS field is to represent text in

Table 1: A comparison between prior, non-LM-based TTS systems against the proposed system. Prior work is mainly based on Mel-spectrogram, Diacritics, and relatively small amounts of training data. We show that while following the LM approach we can leverage large amounts of in-the-wild training data, using plain text, on top of discrete learned speech representations.

| | Prior work | Proposed sys. |
|-----------------------|-----------------|---------------------|
| Intermediate Rep. | Mel spectrogram | Audio discrete rep. |
| Training data (type) | Niqqud | Plain text |
| Training data (hours) | ~ 30 hours | $\sim 5.0k$ hours |

its most basic form, i.e., phonemes [20]. As we aim at building a Diacritic-free system we can not use phonemes as text representations. As a result, we use a sub-word tokenizer in the form of word-piece tokenization. Such tokenizer was found beneficial in text encoders such as BERT [21], and more relevant to our setup AlephBERT [22]. We experimented with several other tokenizers, however, found the word-piece to provide the best overall results (see Section 5 for more details). Below we describe both text tokenization and model in more detail. We depict a general overview of the LM-based approach in Fig. 1.

Text Tokens. We tokenize the text using a word-piece text tokenizer similar to the one proposed by [22]. Specifically, we leverage a pre-trained Hebrew text tokenizer that was trained using 98.7M Hebrew sentences. word-piece tokenizers were tested in different models [21, 23, 24] and performs similarly to Byte-Pair Encoding [25].

Given a training corpus C and a number of desired word-pieces t , the optimization problem is to select t word-pieces such that minimizes the number of word-pieces generated when tokenizing the entire corpus C . We start with a small character vocabulary and special tokens W , and apply merge rules for the elements. iteratively we compute for each pair $w_1, w_2 \in W$ a score as seen in equation 1 and merge the pair with the maximum score getting a new vocabulary $W' = W \cup \{(w_1, w_2)\}$. We follow this step with the new vocabulary until $|W| = t$.

$$score = \frac{freq(e_1, e_2)}{freq(e_1) \times freq(e_2)}, \quad (1)$$

By dividing the frequency of the pair by the product of the frequencies of each of its parts, the algorithm prioritizes the merging of pairs where the individual parts are less frequent in the vocabulary.

Model. Recall, our goal is to produce a Diacritic-free Hebrew TTS system that can handle weakly transcribed spoken data. Hence, we proposed leveraging the abilities of language models to efficiently model long contexts. Inspired by recent LM-based approaches for TTS [8, 18], our model uses an LM approach that operates directly over discrete representation obtained from a pre-trained speech encoder.

The model first receives a text prompt, x_i , and a 3-second enrolled recording as an acoustic prompt. We then, encode the acoustic prompt using the same speech encoder E , and process the text using the text tokenizer defined in sub-section 2. Recall, that the speech encoder, E , quantizes the utterance using RVQ module, hence it outputs a matrix of size $T \times N_{cb}$. Meaning that, at each time step we are left with N_{cb} discrete codes.

There are several alternatives in the literature to handle this complex input structure. For instance, the authors in [14, 26] proposed to predict all codes at each time-step in parallel while introducing a delay pattern to better model the conditional prob-

Table 2: Comparison of the LM based approach to both MMS [3] and Overflow [7]. We report both objective metrics (WER, CER, and speaker similarity), together with two human studies evaluating the naturalness and content preservation in the generated samples. In the human study, we report mean and standard deviations.

| Model | Objective Metrics | | | Human Study | |
|-----------------------|-------------------|------|-------------|----------------------------|----------------------------|
| | WER | CER | Spk. Sim. | Naturalness | Content |
| Reference | 0.07 | 0.03 | 0.97 | 4.68 (± 0.46) | 4.63 (± 0.51) |
| MMS [3] | 0.23 | 0.07 | - | 2.51 (± 1.05) | 2.35 (± 0.77) |
| Overflow [7] | 0.20 | 0.08 | 0.88 | 3.44 (± 1.01) | 3.79 (± 0.77) |
| Ours (seen speaker) | 0.19 | 0.08 | 0.95 | 4.17 (± 0.80) | 4.44 (± 0.68) |
| Ours (unseen speaker) | 0.19 | 0.08 | 0.92 | 4.05 (± 0.75) | 4.48 (± 0.58) |

ability distribution. The authors in [16] proposed flattening the whole sequence (resulting in a N_{cb} times larger sequence) and splitting its modeling across two LMs.

In this work, we follow the approach proposed in [8]. In which, the first codebook, $c_{:,1}$, is modeled in an Auto-Regressive (AR) manner following the standard next token prediction framework. Specifically, we concatenate the word-piece tokens with the first codebook from the acoustic prompt, denoted by $w, c_{\leq t,1}$, to infer the next acoustic token $c_{t,1}$ of the target signal. The rest of the codebooks (2 to N_{cb}), are modeled using a non-autoregressive (NAR) model, where the network is trained to maximize the acoustic tokens likelihood derived from the i -th quantizer codebook, conditioned on the sum of all representations from previous codebooks.

Overall, unlike, prior works which are mainly based on Mel-spectrogram as speech representations, diacritics for text, and relatively small and high-quality amounts of training data. Following the LM approach, allows us to leverage large amounts of in-the-wild recordings, using plain text, and operate on top of discrete learned speech representations. Table 1 summarizes the main differences between the methods.

3. Dataset

We use both the *ivrit.ai* dataset [9] together with an internal dataset. Both datasets consists of ~ 4500 hours of speech gathered from local podcasts (~ 1700 from internal and ~ 2800 from *ivrit.ai*). We integrate this dataset with additional in-house data collection efforts. Our in-house dataset comprises spontaneous dialogues sourced from five podcasts featuring multiple speakers discussing a wide range of topics including economy, politics, sports, culture, science, history, and music, among others. The podcast recordings are full episodes, thus containing lengthy audio tracks and various non-speech elements such as music, environmental sounds, and periods of silence. Such real-world conditions present challenges for model optimization and necessitate preprocessing steps. We apply the same pre-processing pipeline to both *ivrit.ai* dataset and our in-house dataset. The preprocessing we have done is as follows. Initially, we standardize all audio recordings to a consistent 16kHz, mono recordings, using julius¹ python package. Subsequently, we employ a Voice Activity Detection (VAD) model, namely *silero-vad* [27] to perform a voice activity detection and segment the waveforms into sentences, filtering out activated segments with a minimum duration of 1 seconds, separating audio segments by a minimal silence duration of 100ms and padding both sides of the segmented audio

¹<https://github.com/adefossez/julius>

with 30ms of silence. Finally, we automatically transcribe the segmented speech using a pre-trained ASR model, specifically Whisper V2-Large [2]. After preprocessing our data, we are left with ~ 1700 hours of natural dialogues.

4. Experiment Setup

4.1. Implementation details

Our model contains 420M parameters and is trained on 8 NVIDIA A30 24GB GPUs with a total batch size of 144,000 acoustic tokens. We optimize the model using *EDEN* scheduler as used in [28] with a starting learning rate of 5×10^{-2} . We train the AR model for 1.2M steps and the NAR for 200k steps. For the audio tokenizer, we use the officially released pretrained version of EnCodec [12] sampled at 24Khz to generate acoustic tokens². To improve the quality of the generated audio we use the pre-trained Multi Band Diffusion (MBD) vocoder [29]. For tokenization, we use the pretrained word-piece tokenizer of AlephBERT³ with vocabulary size of 52k tokens. We train the model for audio length sequences between 1 – 18 seconds. We sample the 50 most likely tokens using $top_k = 50$ and temperature = 1. We adopt the following public code⁴.

4.2. Evaluation metrics

We evaluate the proposed method considering both objective metrics and human study. We consider several axes: (i) content preservation in the form of Word Error Rate (WER), Character Error Rate (CER), and human study; (ii) speaker similarity using a pre-trained speaker verification model; and (iii) overall quality and naturalness via human study. We describe each of the evaluation metrics below.

WER and CER. We calculated Word Error Rates (WERs) and Character Error Rates (CERs) between the input text and an automatic transcription generated by an ASR system. Specifically, we run this evaluation using 100 randomly sampled text prompts with diacritics from SASPEECH [7] dataset. We remove the diacritics for our model and compare with the transcribed text from Whisper V2-Large [2] model which provides state-of-the-art performance. We normalize the text by removing all punctuation from both original and transcribed text. To improve the robustness of the sampling process, we sample three audio generations for each input prompt and select the one with the best WER w.r.t the input text. To calibrate the results

²<https://github.com/facebookresearch/audiocraft/blob/main/docs/ENCODEC.md>

³<https://github.com/OnlpLab/AlephBERT>

⁴<https://github.com/lifeiteng/vall-e>

with the errors produced by the Whisper model, we additionally calculate WER and CER between the reference and transcribed text of the original recordings.

Speaker similarity. For speaker similarity we measure the cosine similarity between the generated speaker and an enrollment set of five different recordings of the person to identify. To compute the cosine similarity we use a state-of-the-art pre-trained speaker verification model [30]. This similarity measure was found to be beneficial in prior work [31, 8, 32].

Human evaluation. We conduct two different human studies to evaluate the quality of the generated samples. Raters were asked to evaluate the quality of the generated speech considering both generation fidelity and naturalness of the speech signals on a scale between 1 – 5, where 1 has the lowest quality and 5 is the best quality. We evaluate 20 samples from each of the evaluated methods while we enforce at least 15 ratings for each sample. All raters are native Hebrew speakers.

Although the Whisper model reaches state-of-the-art performance, its WER in Hebrew is still ~ 27 . Hence, we additionally ask raters to rate the accuracy between the generated speech and the written text. Same as before raters evaluated the content of the recordings on a scale of 1 – 5, where 1 is the least accurate and 5 has a perfect match. We conduct a human study to evaluate the proposed method against the baseline methods as well as to evaluate the text tokenization method.

4.3. Baseline systems

We compare the proposed method against two baseline systems: (i) Massively Multilingual Speech (MMS) [3] and Overflow [7]. The MMS model is based on a multi-lingual wav2vec2.0 [33] trained on $\sim 500k$ hours from 1, 107 languages, while 25 hours in Hebrew. The Overflow model is based on a neural HMM combined with normalizing flows for describing highly non-Gaussian distribution of the acoustics. This model was trained over 30 hours of single-speaker, high-quality data, obtained from the ‘Hayot-Kiss’ podcast [7]. Both methods are based on predicting Diacritics using an external model. In both methods, we use the official pre-trained models released by the authors and follow exactly their text pre-processing pipelines.

5. Results

We start by evaluating the proposed method against both MMS and Overflow. Results are summarized in Table 2. The proposed method provides superior performance to the evaluated baselines considering both objective metrics and human study. Notice, following the LM approach for Hebrew TTS additionally, allows fast adaptation to new speakers. The proposed method shows minor differences in performance when considering speech and unseen speakers.

Interestingly, when considering WER, CER, and Speaker similarity, the Overflow method provides comparable performance to ours while being superior to the MMS model. The main difference between the methods is reflected in the naturalness of the generated speech. Moreover, it is worth mentioning that although the WER and CER are comparable across all methods (with MMS achieving worse WER and Overflow achieving worse CER), these are based on automatic transcriptions that do not take into account the pronunciation, meaning two different words can be transcribed to the same sequence characters while reflecting completely different pronunciation. However, when investigating the content metric under the human study we observe larger differences.

Table 3: Results for LM trained with chars and word-piece text tokenizers. We report WER, CER, and a Human study for content preservation. We report mean and standard deviations for the human study.

| Tokenizer | Objective | | Human Study |
|------------|-----------|------|--------------------|
| | WER | CER | Content |
| Chars | 0.20 | 0.17 | 2.6 (± 0.81) |
| Word-Piece | 0.18 | 0.07 | 3.8 (± 0.75) |

The effect of the tokenizer. As there is no direct mapping between non-diacritic text to phonemes in Hebrew, it is not clear how one should represent the text for the system. A natural approach would be to use character tokenization (i.e., converting the text into a sequence of characters). Another alternative that gains popularity in textual language models is to use a word-piece tokenizer [34]. In this study, we follow the word-piece tokenizer approach.

To better evaluate the effect of using different tokenization methods for the input text we trained two versions of the proposed method using both chars and word-piece tokenizer. We additionally experimented with contextualized representations obtained from hidden layers of a pre-trained text encoder model, namely AlephBERT [22]. Unfortunately, such text representation performs significantly worse than the other tokenizers, hence we do not report results for it. We measure WER and CER metrics, together with a human study measuring content preservation. Results are presented in Table 3. Results suggest that following the word-piece tokenizer provides superior performance to the character-based alternative. This result is being reflected across all the evaluated metrics, however similarly to the results in Table 2, we observe the larger gap when considering subjective content preservation study.

6. Conclusion

In this work, we demonstrate how language models that operate over discrete speech tokens can act as Diacritics-free Hebrew TTS systems. Due to their naturally contextualized manner, language models can better handle ambiguous pronunciations obtained in the absence of diacritics. We empirically show that following the language modeling approach, trained at scale using weakly transcribed data, yields superior performance to non-contextualized, traditional TTS systems when considering context preservation, naturalness, and similarity to the speaker in the generated samples.

Limitations. As our method is based on auto-regressive LM its inference time is relatively long compared to other TTS systems. Moreover, due to its auto-regressive nature, the duration of the generated speech is determined by the model outputs an end-of-sequence token. Additionally, the model can skip words or invent new ones that did not appear in the text prompt. Although we did not observe such behavior significantly affecting model performance, this lack of controllability imposes another limitation when following the LM approach.

Future work. To advance research in the field, more benchmarks are needed. For future work, we aim to tackle this task by constructing high-quality, large-scale speech data, directly dedicated for synthesis purposes.

Acknowledgements This research work was supported by the Israel Innovation Authority, grant number 78563.

7. References

- [1] L. Campbell, "Ethnologue: Languages of the world," 2008.
- [2] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," 2022.
- [3] V. Pratap, A. Tjandra, B. Shi, P. Tomasello, A. Babu, S. Kundu, A. Elkahky, Z. Ni, A. Vyas, M. Fazel-Zarandi, A. Baevski, Y. Adi, X. Zhang, W.-N. Hsu, A. Conneau, and M. Auli, "Scaling speech technology to 1,000+ languages," 2023.
- [4] S. Izre'el, B. Hary, and G. Rahav, "Designing cosih: the corpus of spoken israeli hebrew," *International Journal of Corpus Linguistics*, vol. 6, no. 2, pp. 171–197, 2001.
- [5] J. Azogui, A. Lerner, and V. Silber-Varod, "The open university of israel map task corpus (matakop)," 2016.
- [6] M. Marmorstein and N. Matalon, "The huji corpus of spoken hebrew: An interaction-oriented design of a corpus," 2022.
- [7] O. Sharoni, R. Shenberg, and E. Cooper, "Saspeech: A hebrew single speaker dataset for text to speech and voice conversion," in *Proc. Interspeech*, 2023.
- [8] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.
- [9] Y. Marmor, K. Misgav, and Y. Lifshitz, "ivrit. ai: A comprehensive dataset of hebrew speech for ai research and development," *arXiv preprint arXiv:2307.08720*, 2023.
- [10] S. Mehta, A. Kirkland, H. Lameris, J. Beskow, E. Szekely, and G. E. Henter, "Overflow: Putting flows on top of neural transducers for better tts," Aug. 2023. [Online]. Available: <http://dx.doi.org/10.21437/Interspeech.2023-1996>
- [11] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in neural information processing systems*, vol. 33, pp. 17 022–17 033, 2020.
- [12] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.
- [13] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2021.
- [14] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [15] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, "Audiogen: Textually guided audio generation," *arXiv preprint arXiv:2209.15352*, 2022.
- [16] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi *et al.*, "Audiolm: a language modeling approach to audio generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.
- [17] R. Sheffer and Y. Adi, "I hear your true colors: Image guided audio generation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [18] E. Kharitonov, D. Vincent, Z. Borsos, R. Marinier, S. Girgin, O. Pietquin, M. Sharifi, M. Tagliasacchi, and N. Zeghidour, "Speak, read and prompt: High-fidelity text-to-speech with minimal supervision," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 1703–1718, 2023.
- [19] D. Lyth and S. King, "Natural language guidance of high-fidelity text-to-speech with synthetic annotations," *arXiv preprint arXiv:2402.01912*, 2024.
- [20] X. Tan, T. Qin, F. Soong, and T.-Y. Liu, "A survey on neural speech synthesis," *arXiv preprint arXiv:2106.15561*, 2021.
- [21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.
- [22] A. Seker, E. Bandel, D. Bareket, I. Brusilovsky, R. S. Greenfeld, and R. Tsarfaty, "Alephbert: A hebrew large pre-trained language model to start-off your hebrew nlp application with," *arXiv preprint arXiv:2104.04052*, 2021.
- [23] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.
- [24] M. Schuster and K. Nakajima, "Japanese and korean voice search," in *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2012, pp. 5149–5152.
- [25] P. Gage, "A new algorithm for data compression," *The C Users Journal archive*, vol. 12, pp. 23–38, 1994. [Online]. Available: <https://api.semanticscholar.org/CorpusID:59804030>
- [26] E. Kharitonov, A. Lee, A. Polyak, Y. Adi, J. Copet, K. Lakhotia, T.-A. Nguyen, M. Rivière, A. Mohamed, E. Dupoux *et al.*, "Text-free prosody-aware generative spoken language modeling," in *ACL 2022-Association for Computational Linguistics*, vol. 1. MIT Press, 2022, pp. 8666–8681.
- [27] S. Team, "Silero vad: pre-trained enterprise-grade voice activity detector (vad), number detector and language classifier," 2021.
- [28] Z. Yao, L. Guo, X. Yang, W. Kang, F. Kuang, Y. Yang, Z. Jin, L. Lin, and D. Povey, "Zipformer: A faster and better encoder for automatic speech recognition," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: <https://openreview.net/forum?id=9WD9KwssyT>
- [29] R. San Roman, Y. Adi, A. Deleforge, R. Serizel, G. Synnaeve, and A. Défossez, "From discrete tokens to high-fidelity audio using multi-band diffusion," *arXiv preprint arXiv:*, 2023.
- [30] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, p. 1505–1518, Oct. 2022. [Online]. Available: <http://dx.doi.org/10.1109/JSTSP.2022.3188113>
- [31] A. Polyak, Y. Adi, J. Copet, E. Kharitonov, K. Lakhotia, W.-N. Hsu, A. Mohamed, and E. Dupoux, "Speech resynthesis from discrete disentangled self-supervised representations," *arXiv preprint arXiv:2104.00355*, 2021.
- [32] C. Wang, W.-N. Hsu, Y. Adi, A. Polyak, A. Lee, P.-J. Chen, J. Gu, and J. Pino, "fairseq s²: A scalable and integrable speech synthesis toolkit," *arXiv preprint arXiv:2109.06912*, 2021.
- [33] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.
- [34] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," *arXiv preprint arXiv:2402.06196*, 2024.