Phonikud: Hebrew Grapheme-to-Phoneme Conversion for Real-Time Text-to-Speech

Yakov Kolani[1] **Maxim Melichov**[2] **Cobi Calev**[1] **Morris Alper**[3]
[1]Independent Researcher [2]Reichman University [3]Tel Aviv University

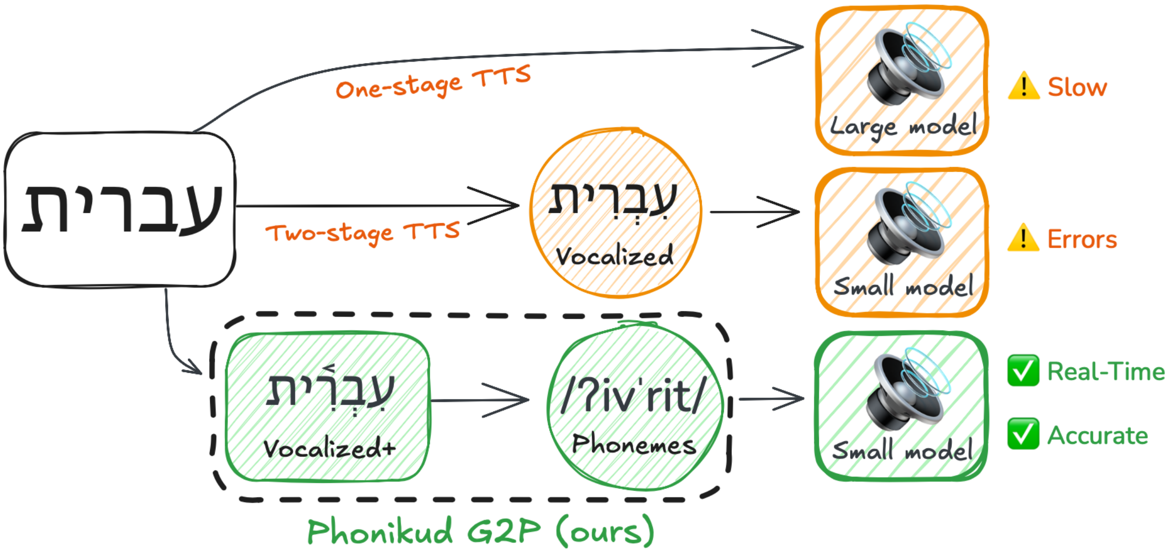

Figure 1: **Phonikud: Hebrew G2P conversion for fast, phonetically accurate Hebrew TTS.** Hebrew writing normally omits vowels, creating a speed-accuracy tradeoff for TTS: large models (orange, top) trained on raw Hebrew text are more phonetically accurate but slow, while small models trained on vocalized text (orange, middle) produce pronunciation errors because added vowel marks still omit critical features like stress placement. Phonikud (green) enables both speed and phonetic accuracy through enhanced vocalization (*Vocalized+* above) augmenting standard vowel marks with additional symbols indicating phonetic features such as stress, followed by conversion to standard IPA phonemes. This enables training small TTS models with phonetically accurate outputs suitable for real-time applications.

Abstract

Real-time text-to-speech (TTS) for Modern Hebrew is challenging due to the language's orthographic complexity. Existing solutions ignore crucial phonetic features such as stress that remain underspecified even when vowel marks are added. To address these limitations, we introduce *Phonikud*, a lightweight, open-source grapheme-to-phoneme (G2P) system that outputs fully-specified IPA transcriptions for Hebrew text. Our approach adapts an existing diacritization model with lightweight adaptors, incurring only negligible additional latency. We also contribute *ILSpeech*, a novel dataset of Hebrew speech with expert-annotated IPA transcriptions that serves both as the first benchmark for Hebrew G2P and as essential training data for TTS systems. Experimental results demonstrate that Phonikud achieves substantially higher accuracy in Hebrew G2P compared to prior methods, and that G2P preprocessing enables training of effective real-time Hebrew TTS models with superior speed-accuracy trade-offs. We release all code, data, and models at https://phonikud-paper.netlify.app to advance Hebrew TTS research.

1 Introduction

Despite the Modern Hebrew language being spoken by approximately nine million people (Lewis, 2009), it currently lacks an open-source real-time text-to-speech (TTS) system with adequate performance. TTS systems for important applications such as screen readers for visually impaired users and for smart home technology must run in real-time on resource-constrained devices. However, applying standard techniques to Hebrew is challenging due to the language's opaque orthography,

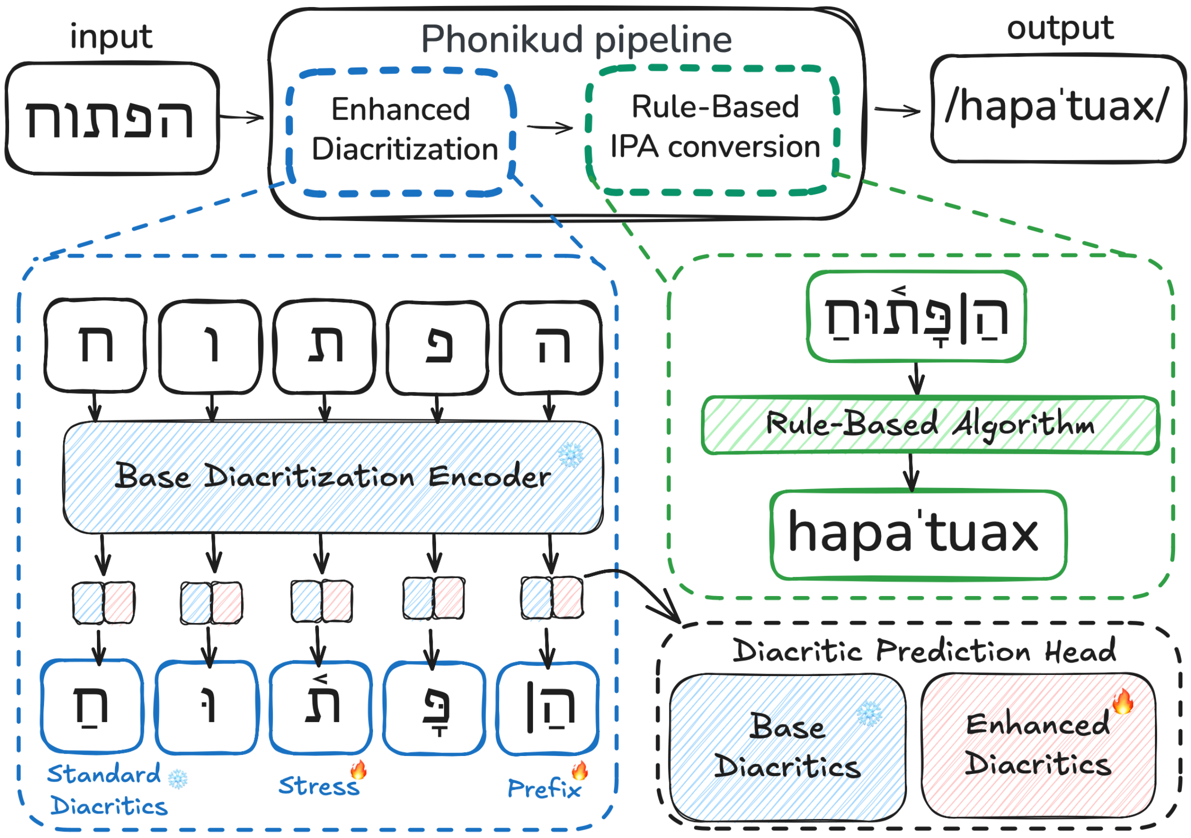

Figure 2: **The Phonikud grapheme-to-phoneme pipeline.** Phonikud converts unvocalized Hebrew text into fully-specified IPA in two steps: First, an enhanced diacritization module adds standard vowel marks and enhanced phonetic symbols to each letter of the text. This is done with a frozen (ice symbol above) base diacritization model, which is a character-level encoder model, and its per-character prediction head for standard vowel diacritics. This is augmented with an additional trainable (fire symbol above) linear adaptor serving as a head for predicting enhanced diacritics. Second, a rule-based transformation module converts this text with enhanced diacritics, which disambiguate its phonetic content, into IPA. This output may be used to train small, real-time TTS models.

which is difficult to parse directly for the small TTS models needed to achieve low latency (as we show in Section 5).

The Hebrew script omits phonetic features such as vowel sounds, leaving them to be inferred from context. For instance, in Hebrew the word ספר may be read as /ˈsefer/ (“book”), /saˈpar/ (“barber”), /saˈfar/ (“he counted”), or /sfar/ (“suburb”). A system of optional diacritics (*nikud*) may be used to indicate these features, but they are mostly confined to pedagogical texts such as dictionaries. Moreover, the pronunciation of a Hebrew word cannot be unambiguously determined even when vowel diacritics are provided. For example, בִּירָה may be read as either /ˈbira/ (“beer”) or /biˈra/ (“capital city”). This *phonetic underspecification* challenges TTS systems, which must receive normal (unvocalized) Hebrew text as input and output correctly-pronounced Hebrew audio.

One approach to Hebrew TTS maps unvocalized Hebrew text directly to audio (Roth et al., 2024; Zeldes et al., 2025). This may achieve high quality given a large model with sufficient capacity to capture the complexities of Hebrew orthography. However, such models are computationally intensive and incur high latency, making them unsuitable for real-time applications. Conversely, small TTS models struggle to predict accurate pronunciation from unvocalized Hebrew. Existing approaches predict vowel diacritics directly (Sharoni et al., 2023; Pratap et al., 2024), but this does not fully resolve ambiguity (as in the example above), leading to inaccurate pronunciations in TTS outputs.

To bridge this gap, we propose a lightweight grapheme-to-phoneme (G2P) pipeline, *Phonikud*, to resolve the phonetic ambiguities in written Hebrew, enabling the training of small, low-latency TTS models for Hebrew speech synthesis.

Specifically, we adapt an existing state-of-the-art (SOTA) model for predicting Hebrew vowel diactitics (Shmidman et al., 2023), adding lightweight adaptors to predict additional phonetic features such as stress and *shva* realization (see Section 2) needed for disambiguation. A rule-based module converts these outputs into the International Phonetic Alphabet (IPA). We show that this allows effectively training small, real-time-capable TTS models.

As an additional contribution to enable our goal, we contribute the novel *ILSpeech* dataset and benchmark, consisting of high-quality Hebrew speech recordings along with Hebrew text and expert-annotated IPA transcriptions. This serves both as an additional training resource for Hebrew TTS, which currently has a dire lack of available open data, as well as a benchmark for evaluating the novel task of G2P for Hebrew text.

In summary, our key contributions are:

- A lightweight, open-source G2P model augmenting an existing Hebrew diacritizer to accurately transcribe Hebrew text in IPA.
- Results demonstrating that G2P is beneficial for training real-time TTS systems for Hebrew, along with comparisons to existing systems.
- *ILSpeech*, a novel dataset and benchmark of Hebrew speech recordings, Hebrew and IPA transcriptions, enabling TTS training and benchmarking Hebrew G2P.

We release[1] our data, code, and trained models to spur development of real-time Hebrew TTS systems.

2 Phonetic Underspecification in Hebrew

Hebrew is normally written without vowel marks (*unvocalized text*), but even when these are added (*vocalized text*) it is still underspecified for various phonetic features that are needed for accurate TTS. These may be split into three primary issues:

Stress. Hebrew has lexical stress, which is only partially predictable from word shape and part of speech (Graf and Ussishkin, 2003). As illustrated by the minimal pair /ˈtxina/ (“tahini”) vs. /txiˈna/ (“grinding”), both spelled טְחִינָה, stress is not indicated in the orthography even when vowel marks are provided.

Shva. The vowel mark known as *shva* is polyvalent, being either silent or pronounced as /e/ depending on context. The latter case occurs subject to complex morpho-phonological rules with many irregularities in loanwords (Weinberg, 1966). For example, in בְּלוֹ֫נְדוֹן /beˈlondon/ (“in London”) the shva vowel between the first two consonants is pronounced, while in בְּלוֹנְדִּ֫ינִי /blonˈdini/ (“blonde”) it is silent.

Irregular words. Infrequently, words may deviate from regular pronunciation rules. A notable case is loanwords containing the phoneme /w/, written identically to /v/. For example, פִּ֫ינְגְּוִין /ˈpingwin/ (“penguin”) is indistinguishable from the hypothetical form */ˈpingvin/. Other examples of irregular spellings include יַ֫אלְלָה /ˈjala/ (“come on”) and יִשָּׂ֫שכָר /jisaˈxar/ (“Issachar”).

These sources of ambiguity motivate our approach of augmenting existing diacritization with additional phonetic features before converting to IPA.

3 Method

Our Phonikud system is illustrated in Figure 2. As a G2P pipeline, Phonikud takes unvocalized Hebrew text as input and outputs fully-specified IPA transcriptions, which can then be used to train efficient Hebrew TTS systems. We proceed to describe the two key components of this system – an enhanced diacritization module (Section 3.1) and rule-based IPA conversion module (Section 3.2) – followed by the novel procedure used for training the system’s learnable components (Section 3.3).

3.1 Enhanced Diacritization

Strong, efficient models have already been developed to add vowel diacritics to Hebrew text, achieving high accuracy on this standard vocalization task (Shmidman et al., 2023). Rather than learning to transcribe from scratch, we leverage this existing capability while extending it to predict additional phonetic features needed for disambiguation. Our key insight is to augment an existing character-level encoder-based diacritization model with lightweight prediction heads to extend the set of symbols which it may predict.

We add three additional symbols that can be predicted for each character position, with logits output by new, trainable MLP prediction heads. We freeze the base encoder model and all existing prediction heads for standard vowel diacritics, while adding and training only the added weights for each new

[1] https://phonikud-paper.netlify.app

enhanced diacritic. This approach offers several advantages: training is extremely lightweight, inference predicts both standard and enhanced diacritics in parallel with minimal runtime overhead compared to the base diacritizer, and performance on standard diacritization remains constant since the base model is frozen.

The three enhanced diacritics we introduce are: (1) a superscript angle indicating a stressed syllable (e.g. לֶ֫חֶם[2]), (2) a subscript line indicating a shva vowel pronounced as /e/ (e.g. מְֽתִיחָה[3]), and (3) a vertical bar indicating the end of a cliticized prefix (e.g. הַ|קוֹד[4]). The first two of these directly indicate missing phonetic features, while the last one aids dictionary matching of irregular words (as they may have prepended clitics). The graphemes used to visually indicate these diacritics include traditional symbols from Biblical texts, which do not appear in ordinary texts.

3.2 Rule-Based IPA Conversion

After generating enhanced vocalized forms (e.g. לֶ֫חֶם), the phonemic representation can be unambiguously determined. We apply a deterministic, rule-based algorithm to convert this to standard IPA (e.g. /ˈlexem/). While TTS models may be trained directly on enhanced Hebrew orthography, we find that IPA conversion significantly benefits model training.

In addition to the advantage of using IPA symbols already recognized by pre-trained TTS models (rather than needing to re-map to the Hebrew alphabet), this also addresses several orthographic complexities of Hebrew:

Many-to-one mappings. Multiple Hebrew graphemes frequently map to a single phoneme. For example, the Hebrew letters ט and ת both represent /t/, and three distinct vowel symbols may map to /e/.

Non-monotonic sequences. Some Hebrew words are parsed non-monotonically (not in a linear order), such as רֵיחַ ("smell") representing /ˈreax/ (not */ˈrexa/, which would be the reading in linear order).

Dual-function letters. The Hebrew letters ו and י may function as vowels or consonants depending on orthographic context. Words such as סִוּוּג /siˈvug/ ("classification") require complex logic to infer that the first occurence of the grapheme ו represents the consonant /v/ while the second coalesces to the vowel /u/.

Irregular words. As discussed in Section 2, irregular words may require dictionary lookup to determine the correct pronunciation.

By addressing these in this IPA conversion stage, Phonikud efficiently simplifies the representation used for TTS training, leading to concrete performance gains.

3.3 Training Procedure

A fundamental challenge in our approach is the lack of existing ground-truth (GT) annotations for Hebrew phonetic features like lexical stress. To address this limitation, we employ a human-in-the-loop procedure to distill knowledge from existing resources along with manual refinement. In particular, we semi-automatically annotate a large-scale Hebrew corpus to indicate stress placement, prefix boundaries, and shva realization. We then distill this knowledge into our model by fine-tuning it on this pseudo-GT.

To produce large-scale data with pseudo-GT annotations, we adopt the IsraParlTweet corpus consisting of 5M sentences of Hebrew text (Mor-Lan et al., 2024). We leverage Dicta's[5] morpho-phonological analysis API which returns stress placement and prefix boundaries for Hebrew words in isolation; applied to each word in the corpus, this produces partial annotations, although the API lacks many words and is frequently inaccurate. As shva realization is partially predictable, we also apply a set of known rules to automatically annotate it within words. As this procedure is frequently inaccurate, we correct many cases of errors via manual expert annotation, by sorting words types by frequency and correcting the most common items.

Our results show that training on this pseudo-GT data leads to superior performance on predicting the phonetic representation of Hebrew texts to existing approaches.

4 ILSpeech

We introduce *ILSpeech*, a high-quality Hebrew speech dataset with expert-annotated phonetic transcriptions. This dataset serves two primary purposes: (1) establishing a benchmark for Hebrew G2P systems by providing ground-truth phonetic transcriptions for systematic evaluation, and (2) supplying high-quality training data for Hebrew TTS

[2] Pronounced /ˈlexem/ ("bread").

[3] Pronounced /metiˈxa/ ("stretch").

[4] Pronounced /haˈkod/ ("the code"), with prefixed /ha-/

[5] https://dicta.org.il

Model	WER	WER$_{-s}$	CER	בוקר טוב
Phonikud (Ours)	0.20	0.15	0.04	ˈboker ˈtov
Diacritizers*				
DictaBERT	0.38	0.24	0.08	boˈker ˈtov
Nakdimon	0.40	0.27	0.09	boˈker ˈtov
Multilingual G2P				
eSpeak NG	1.00	0.96	0.47	vvkr tov
Goruut	1.00	0.95	0.48	boʁe̯ʁ tˤoːβ
CharsiuG2P	1.00	0.99	0.71	boːʔab têːb

Table 1: **G2P evaluation and example.** We calculate G2P performance on our ILSpeech benchmark, using unvocalized Hebrew as input and evaluating error rates relative to ground-truth IPA transcription. WER$_{-s}$ indicates word error rate while disregarding mismatched stress. We also illustrate the output applied to a Hebrew phrase (with GT /ˈboker ˈtov/).

*Using our IPA conversion rules with defaults for ambiguous features like stress.

development. Our dataset contains approximately two hours of studio-quality speech from two native Hebrew speakers (∼1.5K sentences), representing a proof-of-concept that may be extended with additional speakers and content as needed.

ILSpeech provides time-aligned transcriptions with two parallel tiers: (1) unvocalized Hebrew text, and (2) expert-annotated IPA transcriptions. The latter fully specifies phonetic features such as stress that are ambiguous in vocalized Hebrew text. To the best of our knowledge, this is the first open Hebrew audio corpus containing full IPA transcriptions, newly enabling evaluation of Hebrew G2P systems on previously unmeasurable features such as stress placement and shva realization.

The dataset addresses a critical gap in Hebrew speech resources. While several Hebrew audio corpora exist, they lack the phonetic detail necessary for G2P evaluation. In addition, existing corpora are mostly small-scale (Izre'el et al., 2001; Azogui et al., 2016; Marmorstein and Matalon, 2022; Sharoni et al., 2023), while the only open large-scale corpora contain low-quality recordings without proper segmentation (Marmor et al., 2023; Turetzky et al., 2024), unsuitable for high-quality TTS training. As such, ILSpeech provides an important contribution with high-quality recording data along with expert phonetic annotations necessary for rigorous G2P evaluation.

We release ILSpeech under permissive licensing with requirements for ethical use to support open research in Hebrew speech technology.

Model	WER	CER	RTF	# Params
Phonikud (Ours)				
Piper	0.17	0.06	0.09	20M
StyleTTS2	0.13	0.04	0.50	90M
Open Models				
One-Stage:				
MMS	0.23	0.07	0.21	36M
SASPEECH	0.20	0.08	0.16	28M
Robo-Shaul	0.21	0.08	1.58	23M
Two-Stage:				
LoTHM	0.10	0.03	84.75	211M
HebTTS	0.19	0.08	25.44	428M
Proprietary Models				
Google	0.11	0.04	4.08	—
OpenAI	0.11	0.03	1.60	—

Table 2: **TTS Comparison.** Above we measure the performance of TTS models trained using our Phonikud G2P conversion (top), existing open models for Hebrew (middle), and proprietary models available via cloud APIs (bottom). Accuracy metrics (WER, CER) are calculated by applying an ASR system to synthesized audio and comparing the output transcripts to the original input text. Latency (RTF) is measured using a consistent hardware setup for open models, and via external APIs for proprietary models. Our method allows for training models with a superior trade-off between accuracy and run-time.

5 Results

Results of our Phonikud G2P system are shown in Table 1, evaluated relative to the ground-truth IPA annotations in ILSpeech. This tests generalization as these were not present in the G2P training data (Section 3.3). We calculate word- and character error rates (WER, CER); we also calculate WER when disregarding stress (WER$_{-s}$) as this is the a common failure case of alternative methods. As there lack existing G2P systems designed explicitly for Hebrew with fully-specified IPA (e.g. with stress), we compare to the following baselines: Firstly, we concatenate the existing SOTA Hebrew diacritizers DictaBERT (Shmidman et al., 2023) and Nakdimon (Gershuni and Pinter, 2022) with our IPA conversion rules. As these output vocalized text without our enhanced diacritics disambiguating features like stress, we use reasonable defaults such as always predicting final stress (the most common stress pattern in Hebrew). In addition, we compare to existing open-source multilingual G2P libraries which ostensibly support Hebrew: eSpeak NG[6], Goruut[7], and CharisuG2P[8] (Zhu et al., 2022).

[6] https://github.com/espeak-ng/espeak-ng

[7] https://github.com/neurlang/goruut

[8] https://github.com/lingjzhu/CharsiuG2P

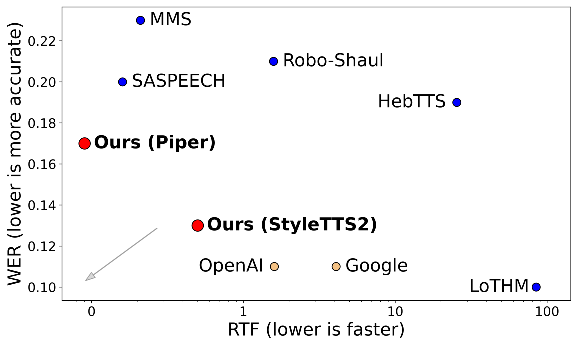

Figure 3: **Speed-accuracy trade-off.** Runtime (x-axis, log-scaled) vs. error rate (y-axis) comparison of our method (red) against open-source (blue) and proprietary (orange) TTS models. The lower-right direction (gray arrow) reflects better overall performance. Our method achieves a superior speed-accuracy trade-off, with our lightest model achieving the best latency and lower error rate than most open models.

Model	WER	CER
Ours	0.26	0.09
-enhanced diacritization	0.27	0.10
-IPA conversion	0.54	0.23
-vowel diacritics	0.58	0.32

Table 3: **Ablation study.** We report performance of downstream TTS training when removing key parts of our full system. Our full system (line 1) is implemented with Piper trained for approximately thirty minutes, for a light-weight comparison. We compare our full system to omitting enhanced diacritization (line 2), further omitting IPA conversion (line 3, i.e. training on vocalized Hebrew text), and to further omitting vowel diacritics (line 4, i.e. training directly on unvocalized Hebrew text). Each ablation leads to further degraded performance.

As seen there, our system outperforms all of these baselines: Our system for disambiguating features such as stress improves performance significantly relative to diacritizers used with our IPA conversion rules, while existing multilingual G2P systems are nearly unusable for Hebrew due to lack of dictionary support for common words (eSpeak NG) and extensive hallucinations in neural models (Goruut, CharsiuG2P). Our pipeline still does does perfectly match the GT IPA annotations in many cases, and we provide an analysis of this in the appendix, showing that these stem from both occasional mistakes in predicting features such as stress, as well from limitations of the base diacritization model on which our model is built (see Section 7.1). In the appendix we also provide results of all of these G2P methods applied to a standard Hebrew text for visual comparison.

We also evaluate our method's utility for downstream TTS by comparing models trained with Phonikud G2P to baseline approaches. In Table 2, we report performance of our method applied to train multiple open-source TTS architectures: a light-weight Piper[9] model, as well as a larger StyleTTS2 (Li et al., 2023) model. We fine-tune existing English TTS checkpoints for these models on audio from ILSpeech along with IPA transcriptions calculated with Phonikud (not using the manual IPA annotations, to fairly evaluate the effect of our G2P pipeline). We compare to various existing Hebrew TTS systems. Among open-source models, MMS (Pratap et al., 2024), the SASPEECH baseline (Sharoni et al., 2023), and Robo-Shaul[10] are light-weight and use a two-stage approach (diacritizing text followed by speech synthesis) while the large models LoTHM (Zeldes et al., 2025) and HebTTS (Roth et al., 2024) take unvocalized Hebrew text as input directly. We also compare to the proprietary TTS models offered by Google[11] and OpenAI [12], which support Hebrew. Following standard practice (Roth et al., 2024), we calculate error rates (WER, CER) by applying an automatic speech recognition (ASR) system to generations and comparing to the original input text. We also report real-time factor (RTF) values to measure latency of each system (including diacritization time, when relevant); for all open systems this is calculated on a consistent, CPU-only hardware setup to match edge computing use cases, while for proprietary models this uses their cloud inference APIs. Following prior work (Roth et al., 2024; Zeldes et al., 2025), we evaluate on a random subset of the SASPEECH (Sharoni et al., 2023) dataset, noting that this is out-of-distribution for our model but in-distribution for the SASPEECH baseline and RoboShaul models.

From these results, we see that our system achieves a superior trade-off between speed and accuracy relative to prior approaches, as our Phonikud G2P system enables training more compact models while preserving phonetic accuracy. We illustrate this trade-off visually in the Pareto frontier analysis in Figure 3. Note that this holds both when comparing to open models run on local hardware, as well as when comparing to proprietary systems run via external API, supporting the value of our method

[9] https://github.com/rhasspy/piper
[10] https://github.com/maxmelichov/Text-To-speech
[11] Gemini 2.5 Flash TTS
[12] GPT-4o mini TTS

for real-time use cases. Qualitative audio comparisons are provided in the supplementary material, illustrating effects such as stress placement that are not directly captured in our automatic, quantitative metrics.

We ablate key parts of our system in Table 3, illustrating the necessity of our full model. All settings use a base Piper model and are trained for a fixed number of iterations (approximately thirty minutes) for a light-weight comparison. Omitting our enhanced diacritics leads to generated speech with incorrect realizations of ambiguous phonetic features such as stress (not fully represented in automatic metrics which are based on ASR results that do not indicate stress placement). Training directly on Hebrew text without our rule-based IPA conversion stage makes the TTS task significantly more challenging, leading to degraded performance with the same number of training steps. This is even more challenging when the Hebrew text is unvocalized, leading the model to incorrectly infer vowels, producing unusable generations. Audio results for our full model and each ablation are provided in the supplementary material.

6 Related Work

Hebrew TTS. Early Hebrew TTS systems relied on rule-based formant synthesis (Laufer, 1975), while modern approaches are mainly learning-based. Some use two-step pipelines that add vowel diacritics before speech synthesis (Sharoni et al., 2023; Pratap et al., 2024), but this fails to resolve key phonetic ambiguities such as lexical stress placement. Others adopt an end-to-end approach, predicting speech directly from raw, undiacritized Hebrew text (Roth et al., 2024; Zeldes et al., 2025), but these require large models with high computational overhead, unsuitable for real-time use. We strike a middle ground by using a two-stage approach for computational efficiency while predicting IPA directly to ensure phonetic accuracy.

G2P conversion. Many languages have opaque orthographies, requiring TTS systems to resolve pronunciation ambiguities. Grapheme-to-phoneme conversion simplifies the learning process for TTS systems by offloading this disambiguation from the text synthesis model (Fong et al., 2019; Hexgrad, 2025). This may handle a variety of language-dependent issues, such as homograph disambiguation in English (e.g. *lead* as a verb vs. noun) (Ploujnikov and Ravanelli, 2022), predicting underspecified tone in Thai (Rugchatjaroen et al., 2019), and inferring vowels in Arabic (Elmallah et al., 2024; Kharsa et al., 2024) and Hebrew (Gershuni and Pinter, 2022; Shmidman et al., 2023). However, existing Hebrew vocalization systems and open-source G2P tools do not specify crucial phonetic features such as stress, while our method generates fully-specified phonetic transcriptions.

7 Conclusion

We have presented a new open-source Hebrew G2P system, Phonikud, and have shown that it newly enables the training of small, real-time Hebrew TTS models, which are needed for edge computing applications. Our experiments show our system compare favorably to existing solutions in both quality and runtime performance. We have also introduced the novel ILSpeech dataset and benchmark for Hebrew G2P evaluation and TTS training. We release our data, code, and trained models to enable applications and research. We envision future work building upon our contributions to further improve Hebrew TTS performance while retaining low latency. Additional promising directions include fine-grained prosody control, support for code-switching, extensive logic for expanding symbols such as dates and addresses, semi-automated IPA annotation to increase the scale of ILSpeech, and extensions to other languages facing related phonological and orthographic challenges.

7.1 Limitations

As our method builds on existing models, we inherit a number of their limitations. The Hebrew diacritization model may output inaccurate productions, leading to incorrect IPA transcriptions. It does not support user selection among alternative vocalizations, which may be desirable in ambiguous cases. The diacritization model adheres to the conventions of formal written Hebrew, which may diverge from spoken norms (e.g. formal /sigˈri/ vs. informal /sgeˈri/ for סגרי "close! (f.)"). Finally, when using our full pipeline, the prosodic quality of synthesized voice is constrained by the inherent trade-offs of real-time TTS models due to their limited capacity.

Ethics Statement

TTS is a dual-use technology: it enables valuable applications such as assistive tools for visually impaired users, but can also misused to generate disinformation or low-quality synthetic content. As with other generative models, responsible use is essential. We believe our work represents an important step towards making language technologies more accessible for lower-resourced languages such as Hebrew, while also acknowledging current limitations in representation. Our proposed dataset, like prior resources for Hebrew, cover a narrow set of speakers and styles, lacking adequate coverage of sociolinguistic variation such as the Mizrahi Hebrew accent. We anticipate that future work will increase this coverage to support more equitable and inclusive voice technologies.

Acknowledgements

We thank Dicta for encouraging our work and for approving our release of our model and data which incorporate their results. We also thank the speakers in ILSpeech for providing an essential resource for the development of Hebrew language technologies.

References

Jacob Azogui, Anat Lerner, and Vered Silber-Varod. 2016. The open university of israel map task corpus (matacop).

Muhammad Morsy Elmallah, Mahmoud Reda, Kareem Darwish, Abdelrahman El-Sheikh, Ashraf Hatim Elneima, Murtadha Aljubran, Nouf Alsaeed, Reem Mohammed, and Mohamed Al-Badrashiny. 2024. Arabic diacritization using morphologically informed character-level model. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1446–1454.

Jason Fong, Jason Taylor, Korin Richmond, and Simon King. 2019. A comparison between letters and phones as input to sequence-to-sequence models for speech synthesis. In *The 10th ISCA Speech Synthesis Workshop*, pages 223–227. International Speech Communication Association.

Elazar Gershuni and Yuval Pinter. 2022. Restoring hebrew diacritics without a dictionary. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1010–1018.

Dafna Graf and Adam Ussishkin. 2003. Emergent iambs: stress in modern hebrew. *Lingua*, 113(3):239–270.

Hexgrad. 2025. G2p shrinks speech models.

Shlomo Izre'el, Benjamin Hary, and Giora Rahav. 2001. Designing cosih: the corpus of spoken israeli hebrew. *International Journal of Corpus Linguistics*, 6(2):171–197.

Ruba Kharsa, Ashraf Elnagar, and Sane Yagi. 2024. Bert-based arabic diacritization: A state-of-the-art approach for improving text accuracy and pronunciation. *Expert Systems with Applications*, 248:123416.

Asher Laufer. 1975. A programme for synthesizing hebrew speech. *Phonetica*, 32(4):292–299.

M. Paul Lewis, editor. 2009. *Ethnologue: Languages of the World*, sixteenth edition. SIL International, Dallas, TX, USA.

Yinghao Aaron Li, Cong Han, Vinay Raghavan, Gavin Mischler, and Nima Mesgarani. 2023. Styletts 2: Towards human-level text-to-speech through style diffusion and adversarial training with large speech language models. *Advances in Neural Information Processing Systems*, 36:19594–19621.

Yanir Marmor, Kinneret Misgav, and Yair Lifshitz. 2023. ivrit. ai: A comprehensive dataset of hebrew speech for ai research and development. *arXiv preprint arXiv:2307.08720*.

Michal Marmorstein and Nadav Matalon. 2022. The huji corpus of spoken hebrew: An interaction-oriented design of a corpus.

Guy Mor-Lan, Effi Levi, Tamir Sheafer, and Shaul R Shenhav. 2024. Israparltweet: The israeli parliamentary and twitter resource. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9372–9381.

Artem Ploujnikov and Mirco Ravanelli. 2022. Soundchoice: Grapheme-to-phoneme models with semantic disambiguation.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, and 1 others. 2024. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*, 25(97):1–52.

Amit Roth, Arnon Turetzky, and Yossi Adi. 2024. A language modeling approach to diacritic-free hebrew tts. In *Proc. Interspeech 2024*, pages 2775–2779.

Anocha Rugchatjaroen, Sittipong Saychum, Sarawoot Kongyoung, Patcharika Chootrakool, Sawit Kasuriya, and Chai Wutiwiwatchai. 2019. Efficient two-stage processing for joint sequence model-based thai grapheme-to-phoneme conversion. *Speech Communication*, 106:105–111.

Orian Sharoni, Roee Shenberg, and Erica Cooper. 2023. Saspeech: A hebrew single speaker dataset for text to speech and voice conversion. In *Proc. Interspeech*.

Shaltiel Shmidman, Avi Shmidman, and Moshe Koppel. 2023. Dictabert: A state-of-the-art bert suite for modern hebrew. *Preprint*, arXiv:2308.16687.

Arnon Turetzky, Or Tal, Yael Segal, Yehoshua Dissen, Ella Zeldes, Amit Roth, Eyal Cohen, Yosi Shrem, Bronya R Chernyak, Olga Seleznova, and 1 others. 2024. Hebdb: a weakly supervised dataset for hebrew speech processing. In *Proc. Interspeech 2024*, pages 1360–1364.

Werner Weinberg. 1966. Spoken israeli hebrew: Trends in the departures from classical phonology. *Journal of Semitic Studies*, 11(1):40–68.

Ella Zeldes, Or Tal, and Yossi Adi. 2025. Enhancing tts stability in hebrew using discrete semantic units. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Jian Zhu, Cong Zhang, and David Jurgens. 2022. Byt5 model for massively multilingual grapheme-to-phoneme conversion. *arXiv preprint arXiv:2204.03067*.