Dict-TTS: Learning to Pronounce with Prior Dictionary Knowledge for Text-to-Speech

Ziyue Jiang*
Zhejiang University
ziyuejiang@zju.edu.cn

Zhe Su*
Zhejiang University
suzhesz00@gmail.com

Zhou Zhao†
Zhejiang University
zhaozhou@zju.edu.cn

Qian Yang
Zhejiang University
qyang1021@foxmail.com

Yi Ren
Sea AI Lab
renyi@sea.com

Jinglin Liu
Zhejiang University
jinglinliu@zju.edu.cn

Zhenhui Ye
Zhejiang University
zhenhuiye@zju.edu.cn

Abstract

Polyphone disambiguation aims to capture accurate pronunciation knowledge from natural text sequences for reliable Text-to-speech (TTS) systems. However, previous approaches require substantial annotated training data and additional efforts from language experts, making it difficult to extend high-quality neural TTS systems to out-of-domain daily conversations and countless languages worldwide. This paper tackles the polyphone disambiguation problem from a concise and novel perspective: we propose Dict-TTS, a semantic-aware generative text-to-speech model with an online website dictionary (the existing prior information in the natural language). Specifically, we design a semantics-to-pronunciation attention (S2PA) module to match the semantic patterns between the input text sequence and the prior semantics in the dictionary and obtain the corresponding pronunciations; The S2PA module can be easily trained with the end-to-end TTS model without any annotated phoneme labels. Experimental results in three languages show that our model outperforms several strong baseline models in terms of pronunciation accuracy and improves the prosody modeling of TTS systems. Further extensive analyses demonstrate that each design in Dict-TTS is effective. The code is available at https://github.com/Zain-Jiang/Dict-TTS.

1 Introduction

Capturing the pronunciations from raw texts is challenging for end-to-end text-to-speech (TTS) systems [2, 31, 42, 45, 55, 27, 14, 41, 22, 21], since there are full of words that are not covered by general pronunciation rules [4, 24, 50]. Therefore, polyphone[3] disambiguation (one of the biggest challenges in converting texts into phonemes [37, 60, 46]) plays an important role in the construction of high-quality neural TTS systems [18, 38]. However, since the exact pronunciation of a polyphone must be inferred based on its semantic contexts, current solutions still face several challenges: 1) the rule-based approaches [64, 19] with limited linguistic knowledge or the neural models [60, 38, 46]

*Equal contribution.
†Corresponding author
[3]Polyphones are characters having more than one phonetic value. See Appendix D for further details.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Character	Pronunciation	Definitions	Usages
乐:			
<l è>	欢喜，快活；使人快乐的事情...	快乐。乐融融。其乐无穷。乐观。乐天。取乐。逗乐。快乐...	
<y uè>	声音，成调的声音。或姓氏...	音乐。声乐。乐池。乐音。乐曲（①音乐与歌曲；②伴奏...	
<y ào>	喜好、欣赏。用于文言文...	知者乐水，仁者乐山。	
<l ào>	地名用字。	河北省乐亭、山东省乐陵。	

Figure 1: The illustration of the dictionary entry that contains information on character's or word's definitions, usages, and pronunciations. For example, in the Chinese sentence "快乐的人们", the pronunciation of the polyphone "乐" should be inferred based on its semantic contexts.

trained on limited data suffer from significant performance degradation on out-of-domain text datasets; 2) the neural network-based models [6, 5, 38] learn the grapheme-to-phoneme (G2P) mapping in an end-to-end manner without explicit semantics modeling, which hinders their pronunciation accuracy in real-life applications. 3) based on the above two points, a reliable polyphone disambiguation module is usually based on a combination of hand-crafted rules, structured G2P-oriented lexicons, and neural models [16], which requires substantial phonemes labels and external knowledge from language experts.

Unlike the previous rule-based or neural network-based approaches, we address the above challenges with the existing prior information worldwide. As shown in Figure 1, an arbitrary dictionary used in daily life can be viewed as a prior knowledge database. Intuitively, it contains valuable prior knowledge for pronunciations in conversations. When one is confused about the acoustic pronunciation of a specific polyphone, he or she will resort to the dictionary website to infer its exact reading based on the semantic context. We imitate this scenario in our architecture design and propose Dict-TTS, an unsupervised polyphone disambiguation framework, which explicitly consults the online dictionary to identify the correct semantic meanings and acoustic pronunciations of polyphones. Specifically,

- To explicitly learn the semantics-to-pronunciation mapping, we adopt a semantic encoder to obtain the semantic contexts of the input text and utilize a semantics-to-pronunciation attention (S2PA) module to search the matched semantic patterns in the dictionary so as to find the correct pronunciations. We also use the retrieved semantic information as auxiliary information for prosody modeling of the TTS model.
- To perform polyphone disambiguation without phoneme labels, we combine our S2PA module into end-to-end TTS systems' training and inference processes. Different from current neural polyphone disambiguation models, our module can be trained with the guidance of mel-spectrogram reconstruction loss in a fully end-to-end manner, which significantly reduces the cost of building such a system.

To demonstrate the generalization ability of our Dict-TTS, we perform experiments on three datasets, including a standard Mandarin dataset [3], a Japanese corpus [47], and a Cantonese dataset [1]. Experiments on these datasets show that Dict-TTS outperforms other state-of-the-art polyphone disambiguation models in pronunciation accuracy and improves the prosody modeling of TTS systems in terms of both subjective and objective evaluation metrics. The pronunciation accuracy of Dict-TTS is further improved by being pre-trained on a large-scaled automatic speech recognition (ASR) dataset. The main contributions of this work are summarized as follows:

- We incorporate the online dictionary into TTS systems and propose a semantic-aware method for polyphone disambiguation, which improves the pronunciation accuracy and robustness of end-to-end TTS systems. Moreover, the idea of introducing the prior knowledge worldwide can also inspire other tasks like neural language modeling [26] and sequence labeling [34].
- We propose a novel and general framework for unsupervised polyphone disambiguation in TTS systems, which further enables the efficient pre-training on large-scaled ASR datasets and improves the generalization capacity significantly.
- We also find that the retrieved semantics in the dictionary knowledge can be used as auxiliary information to improve prosody modeling and help the TTS system to generate more expressive speech.

- We further analyze the characteristics of the linguistic encoder based on phoneme and character and provide valuable interpretations about our semantics-to-pronunciation attention module.

2 Background

This section describes the background of TTS, grapheme-to-phoneme (G2P) pipeline, and their relations with polyphone disambiguation. We also review the existing works that aim at semantic-aware polyphone disambiguation and analyze their advantages and disadvantages.

Text-to-Speech Text-to-speech (TTS) models [55, 2, 30, 31, 42, 27, 41, 32] first generate mel-spectrogram from text and then synthesize speech waveform from the generated mel-spectrogram using a separately pre-trained vocoder [52, 29, 20], or directly generate waveform from text in an end-to-end manner [40, 13, 28]. The frontend model of end-to-end TTS system should tackle one important task, i.e., polyphone disambiguation [65]. Properly mapping the grapheme sequence into phoneme sequence requires the linguistic encoder to capture the empirical pronunciation rules in daily conversations. However, it is extremely difficult for the linguistic encoder to learn all the pronunciation rules necessary to produce speech in an end-to-end manner, which results in inevitable mispronunciations in the generated speech. To alleviate this problem, robust TTS models usually convert the text sequence into the phoneme sequence with open-source grapheme-to-phoneme pipelines and predict the mel-spectrogram from the phoneme sequence. However, the rule-based or neural network-based grapheme-to-phoneme pipelines suffer from significant performance degradation on out-of-domain datasets since it is extremely difficult and costly for them to cover all linguistic knowledge.

Grapheme-to-Phoneme The grapheme-to-phoneme models map grapheme sequence to phoneme sequence to reduce pronunciation errors in modern TTS systems. For logographic languages like Chinese, Japanese and Korean, although the lexicon can cover nearly all the characters, there are full of polyphones that can only be decided according to the semantic context of a character [50]. Thus, polyphone disambiguation is the most important challenge in grapheme-to-phoneme conversions for this kind of languages. Moreover, many alphabetic languages including English and French also have polyphones in daily conversations. Current polyphone disambiguation approaches can be categorized into the rule-based approach [64, 19] and the data-driven approach [43, 6, 5, 38]. The rule-based G2P method is based on a combination of hand-crafted rules and structured G2P-oriented lexicons, which requires a substantial amount of linguistic knowledge. The data-driven G2P algorithm adopts statistical methods [33] or neural encoder-decoder architecture [58, 5, 48, 60, 38, 46, 54]. However, building a data-driven G2P model requires a large amount of carefully labeled data and substantial linguistic knowledge from language experts, which is extremely costly and laborious.

Semantic-Aware Polyphone Disambiguation Polyphone disambiguation is the core issue for G2P conversion in various languages. The pronunciation of a polyphone is defined by the semantic context information of neighbouring characters [50]. In order to comprehend the semantic meaning in the given sentence for polyphone disambiguation, previous methods [11, 57, 49, 18, 8] have adopted the pre-trained language model [12] to extract semantic features from raw character sequences and predict the pronunciation of polyphones with neural classifiers according to the semantic features. Among them, PnG BERT and Mixed-Phoneme BERT [25, 62] take both phoneme and grapheme as input to train an augmented BERT and use the pre-trained augmented BERT as the TTS encoder. However, these methods still require annotated data to train and can not be incorporated into the TTS training in an end-to-end manner. Although NLR [16] directly injects BERT-derived knowledge into the TTS systems without phoneme labels and successfully reduces pronunciation errors, their method confounds the acoustic and semantic space, which significantly affects the pronunciation accuracy.

3 Method

To exploit the prior linguistic knowledge in the online dictionary for TTS systems, we propose Dict-TTS, which explicitly captures the semantic relevance between the input sentences and the dictionary entries for polyphone disambiguation. In this section, we firstly introduce the overall architecture design of Dict-TTS based on PortaSpeech [41]. Then after the comparison between the

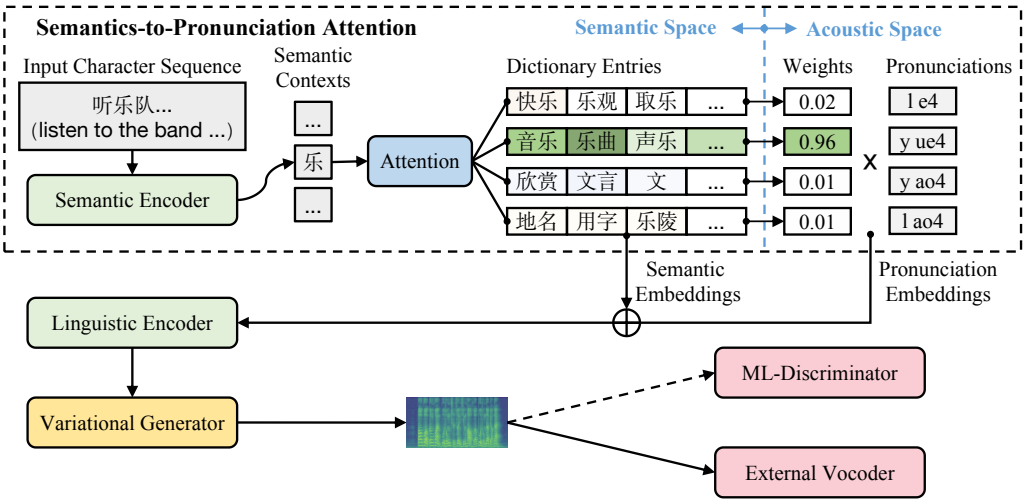


Figure 2: The overall architecture for Dict-TTS. The character "乐" has 4 possible pronunciations coupled with 4 different meanings. The module inside the dashed box is our semantics-to-pronunciation attention (S2PA). The S2PA module measures the semantic similarity between the input character and the corresponding dictionary entries and aggregates the attention weights into pronunciation weights. Then the weighted semantic embeddings and pronunciation embeddings are fed into the linguistic encoder for feature fusion. The semantic and acoustic spaces described in Subsection 3.2 are separated by the blue dashed line. "ML-Discriminator" denotes Multi-Length Discriminator in HiFiSinger [7]. The dashed black line denotes that the operation is only executed in the training phase.

phoneme-based and character-based TTS systems in the acoustic and semantic space, we design a novel semantics-to-pronunciation attention (S2PA) module to learn grapheme-to-phoneme mappings based on the semantic context; In general, Dict-TTS exploits the prior pronunciation knowledge in the online dictionary with the following steps: Firstly, the character sequence is fed into the self-attention based semantic encoder to obtain the semantic representations of the input character sequence, and we utilize a pre-trained cross-lingual language model [10] to extract the semantic context information in the dictionary entries; Secondly, we calculate the most relevant dictionary entries of the input graphemes and obtain the corresponding pronunciation sequence contained in the dictionary entries; Finally, the extracted semantic context information and the pronunciations are fed into the linguistic encoder for feature fusion. We describe these designs in detail in the following subsections.

3.1 Model Overview

The overall model architecture of Dict-TTS is shown in Figure 2. Dict-TTS keeps the main structures of PortaSpeech: a Transformer-based linguistic encoder; a VAE-based variational generator with flow-based prior to generate diverse mel-spectrogram. The flow-based post-net in PortaSpeech is replaced with a multi-length discriminator [7] based on random windows of different lengths, which has been proved to improve the naturalness of word pronunciations [59]. However, since there are no phoneme inputs in our scenarios, we replace the linguistic encoder that combines hard word-level alignment and soft phoneme-level alignment with: 1) a semantic encoder that extracts the semantic representations in the grapheme sequence; 2) a semantics-to-pronunciation attention module that matches the semantic patterns between the dictionary entries and the grapheme representations and obtains the corresponding semantic embedding and pronunciation embedding; 3) a linguistic encoder that fuses the semantic embedding and pronunciation embedding.

3.2 Comparison between the phoneme-based and character-based TTS systems

In this subsection, we make preliminary analyses about the linguistic encoder of phoneme-based and character-based TTS systems. For simplicity, we describe the following concepts according to the logographic writing system, where a single written character represents a complete grammatical word

or morpheme. These concepts can be extended to alphabetic languages like English by replacing "character" with "word".

Phoneme-based TTS systems As is shown in Figure 3, the linguistic encoder of the phoneme-based TTS system takes phoneme sequence p generated by the G2P module as inputs. The main challenge of the phoneme-based linguistic encoder is to comprehend the semantic and syntactic representation s from p and deduce the pitch trajectory, speaking duration, and other acoustic features from s and p to generate the expressive and natural pronunciation hidden state g. Since the phoneme sequence p is a combination of the smallest units of sound in speech, it may be ambiguous in terms of semantic meaning, which brings difficulties for the deduction of the representation s. For example, "AE1, T, F, ER1, S, T" can be easily classified as "At first", but "W, EH1, DH, ER0" can be classified as "Whether" or "Weather". Homophones like "to", "too", and "two" can be converted to the same phoneme sequence "T, UW1", but their local speaking duration and pitch are different. Moreover, the tree-structured syntactic information contained in the word-based input sequence is missing. Since semantic information and syntactic information possess rich intonational features such as pitch accent and phrasing of the input text [59], the ambiguity of s hurts the prosody modeling in phoneme-based TTS systems.

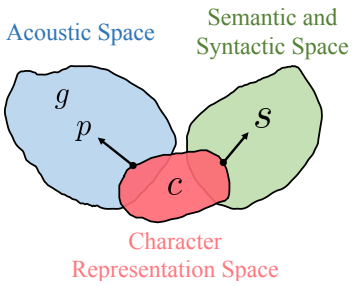


Figure 3: The illustration of representations in the linguistic encoder.

Character-based TTS systems The first challenge for a character-based TTS system is to predict the correct phoneme sequence p. Unlike the phoneme-based TTS system, the character-based TTS system does not know the phoneme sequence p when characters arrive. Thus, it does not know how to pronounce the words accurately at first. Then the mel-spectrogram reconstruction loss in TTS training would drag the character representation c to the acoustic space. For example, Chinese characters "火" ("fire" in English) and "伙" ("partner" in English) share the same pronunciation ("H UO3") and have different semantic meanings. However, with the guidance of the mel-spectrogram reconstruction loss, their representations distribute according to the acoustic pronunciation, which hinders the semantic comprehension for polyphone disambiguation and prosody modeling.

From the above analyses, it can be seen that the character representations c should locate in the semantic space so that we can easily capture s based on the context, deduce p based on the dictionary and s, and finally obtain the natural pronunciation hidden state g. The following subsection mainly describes how we achieve the above goals with our semantics-to-pronunciation attention module.

3.3 Semantics-to-Pronunciation Attention

As shown in Figure 2, the semantics-to-pronunciation attention (S2PA) module is designed for explicit semantics comprehension and polyphone disambiguation.

Dictionary Definition Assuming that we have an dictionary D which contains a sequence of characters $C = [c_1, c_2, ..., c_n]$, where n is the size of characters set in a language[4]. In the dictionary, each character c_i has a sequence of possible pronunciations $p_i = [p_{i,1}, p_{i,2}, ..., p_{i,m}]$ and each pronunciation $p_{i,j}$ has its corresponding dictionary entry $e_{i,j} = [e_{i,j,1}, e_{i,j,2}, ..., e_{i,j,u}] \in E$ (definitions, usages, and translations are merged together as a single characters sequence), where m is the number of the possible pronunciation of c_i and u is the number of characters in the corresponding entry. Note that for polyphones $m > 1$ and for characters that have only one pronunciation $m = 1$.

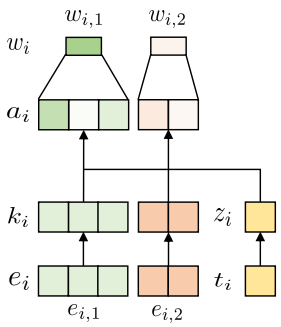


Figure 4: The illustration of semantic pattern matching in S2PA module.

[4]The dictionary can be easily downloaded from online websites. See Appendix A.1 for further details

Semantics Matching The goal of our S2PA is to obtain the pronunciation sequence p by measuring the semantic similarity between the input character sequence $t = [t_1, ..., t_l]$ and the corresponding gloss items in the dictionary, where l is the sequence length. As shown in Figure 4, we firstly extract the semantic context information $\mathbf{k}$ of every entry e with a pretrained cross-lingual language model [10] and store $\mathbf{k}$ as the prior dictionary knowledge. Then we use a semantic encoder to obtain the semantic contexts $\mathbf{z}$ from the input character sequence t. For each character token t_i, its semantic feature vector $\mathbf{z}_i$ is used as the query vector to a semantics-based attention module. Here, attention is used to learn a similarity measure between the semantic feature vector $\mathbf{z}_i$ and $\mathbf{k}_{i,j,k}$:

$$[\mathbf{a}_{i,1,1}, ..., \mathbf{a}_{i,m,u}] = \frac{[\mathbf{k}_{i,1,1}, ..., \mathbf{k}_{i,m,u}] \cdot \mathbf{z}_i^\top}{d} , \tag{1}$$

where d is the scaling factor and $[\mathbf{a}_{i,1,1}, ..., \mathbf{a}_{i,m,u}]$ denotes the semantic similarity between t_i and each item in $[e_{i,1,1}, e_{i,m,u}]$. The retrieved semantic embeddings $\mathbf{s'}_i$ can be extracted by $\mathbf{s'}_i = softmax([\mathbf{a}_{i,1,1}, ..., \mathbf{a}_{i,m,u}]) \cdot [\mathbf{k}_{i,1,1}, ..., \mathbf{k}_{i,m,u}]$. The rich linguistic information in $\mathbf{s'}_i$ can be used as auxiliary information to improve the naturalness and expressiveness of the generated speeches.

Polyphone Disambiguation The aggregated attention weight $\mathbf{w}_{i,j} = \sum_{k=1}^{u} \mathbf{a}_{i,j,k}$ can be seen as the probability of the pronunciation $p_{i,j}$. However, since a polyphone in a specific sentence has only one correct pronunciation, here we use the Gumbel-Softmax function [23] to sample a differentiable approximation of the most possible pronunciation p'_i:

$$w_{i,j} = \frac{\exp\left(\left(\log\left(\mathbf{w}_{i,j}\right) + g_{i,j}\right)/\tau\right)}{\sum_{l=1}^{m} \exp\left(\left(\log\left(\mathbf{w}_{i,l}\right) + g_{i,l}\right)/\tau\right)} , \tag{2}$$

$$p'_i = \sum_{j=1}^{m} w_{i,j} \cdot p_{i,j} , \tag{3}$$

where $g_{i,1}, ...,_{i,m}$ are i.i.d samples drawn from Gumbel(0,1) distribution and τ is the softmax temperature. Then the retrieved pronunciation embeddings p'_i and semantic embeddings $\mathbf{s'}_i$ are then fed into the rest of the linguistic encoder for feature fusion and syntax prediction. Intuitively, our S2PA module can be thought of as an end-to-end method for decomposing the character representation, pronunciation, and semantics with dictionary following the preliminary analyses in Subsection 3.2. Through our S2PA module, the character representation is successfully distributed in the semantic space so that the model can easily deduce the correct pronunciation and semantics based on the lexicon knowledge like human brain.

3.4 Training and Pre-training

In training, the S2PA module weights (including character token embeddings and the pronunciation token embeddings) are jointly trained by the reconstruction loss from the TTS decoder. Thus, our Dict-TTS does not require any explicit phoneme labels. In inference, the pronunciations can be specified by feeding the text sequence into the S2PA module to get the predicted pronunciations. Besides, our method is compatible with the predefined rules from language experts by directly adding specific rules to pronunciation weight $w_{i,j}$.

Although the semantics-to-pronunciation mappings can be explicitly learned by the S2PA module, it could be not accurate enough, due to the following reason: the text training data is not large enough (about 10k sentences) in the TTS dataset, leading to relatively inaccurate context comprehension. To improve the semantic comprehension and the generalization capacity for our S2PA module, we propose a pre-training method using low-quality text-speech pairs from large-scaled automatic speech recognition (ASR) datasets. Since our S2PA module can be trained without hand-crafted phoneme labels, it can be easily pre-trained and effectively finetuned to various domains to improve the pronunciation accuracy of the TTS systems.

4 Experiments

4.1 Experimental Setup

Datasets We evaluate Dict-TTS on three datasets of different sizes, including: 1) Biaobei [3], a Chinese speech corpus consisting of 10,000 sentences (about 12 hours) from a Chinese speaker;

Table 1: The objective and subjective pronunciation accuracy comparisons. PER-O denotes phoneme error rate in the objective evaluation, PER-S denotes phoneme error rate in the subjective evaluation and SER-S denotes sentence error rate in the subjective evaluation.

Method	Biaobei			JSUT			Common Voice (HK)		
	PER-O	PER-S	SER-S	PER-O	PER-S	SER-S	PER-O	PER-S	SER-S
Character	-	3.73%	30.50%	-	13.78%	65.50%	-	1.89%	15.50%
Phoneme	2.78%	1.14%	7.00%	**1.55%**	**0.92%**	**4.25%**	-	1.45%	10.25%
Dict-TTS	**2.12%**	**1.08%**	**6.50%**	3.73%	2.57%	22.75%	-	**1.23%**	**9.75%**

2) JSUT [47], a Japanese speech corpus containing reading-style speeches from a Japanese female speaker (we use the basic5000 subset that contains 5,000 daily-use sentences). 3) Common Voice (HK) [1], a Cantonese speech corpus that contains 125 hours of speeches from 2,869 speakers (We use the 102 hours of the validated speeches). For each of the three datasets, we randomly sample 400 samples for validation and 400 samples for testing. We randomly choose 50 samples in the test set for subjective audio quality and prosody evaluation and use all testing samples for other evaluations. The ground truth mel-spectrograms are generated from the raw waveform with the frame size 1024 and the hop size 256. For computational efficiency, we firstly use the pre-trained XLM-R [10] model to extract the semantic representations from the raw text of the whole dictionary and record them in the disk. We then load the mini-batch along with the pre-constructed dictionary representation during training and testing.

Implementation Details Our Dict-TTS consists of a S2PA module, an encoder, a variational generator and a post-net. The encoder consists of multiple feed-forward Transformer blocks [42] with relative position encoding [44] following Glow-TTS [27]. The encoder and decoder in variational generator are 2D-convolution networks following PortaSpeech [41]. We replace the post-net in PortaSpeech with a multi-length discriminator [7], which has been proved to improve the naturalness of word pronunciations [59]. We add more detailed model configurations in Appendix B.1, B.2. We train Dict-TTS on 1 NVIDIA 3080Ti GPU, with batch size of 40 sentences on each GPU. We use the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$ and follow the same learning rate schedule in [53]. The softmax temperature τ is initialized and annealed using the schedule in [23]. It takes 320k steps for training until convergence. The predicted mel-spectrograms are transformed into audio samples using pre-trained HiFi-GAN [29][5].

4.2 Results of Pronunciation Accuracy

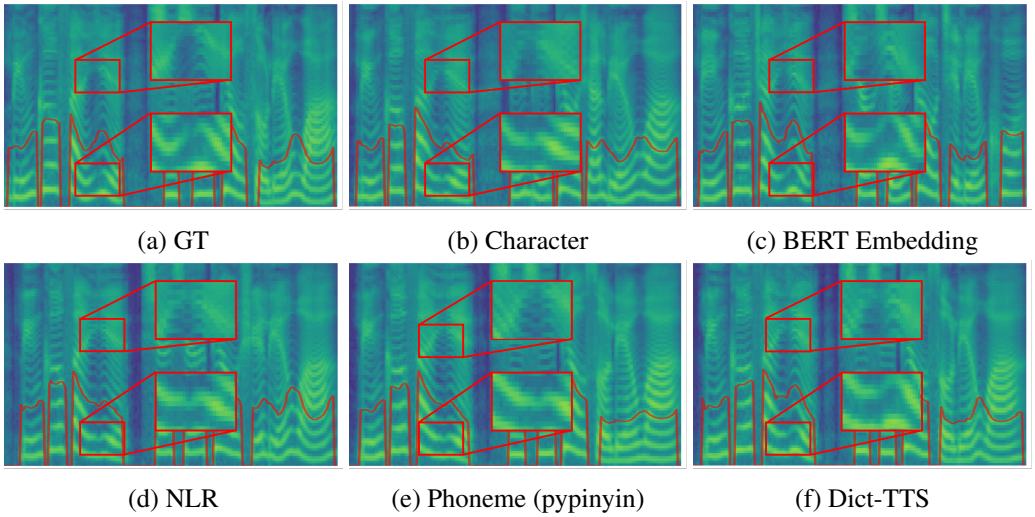
We compare the pronunciation accuracy of our Dict-TTS with other systems, including 1) character-based systems, where we directly feed character into the linguistic encoder; 2) Phoneme-based systems, where we convert the text sequence to the phoneme sequence [55, 45, 27] with most popular open-source grapheme-to-phoneme tools[6]. We measure objective phoneme error rate (PER-O), subjective phoneme error rate (PER-S), and subjective sentence error rate (SER-S) in the evaluations. The phoneme labels in the objective PER evaluation are from the corresponding dataset (since the Common Voice (HK) dataset does not have phoneme labels, we only evaluate the subjective metrics). In the subjective evaluations, each audio in the test set is listened by at least 4 language experts. We ask them to write down the mispronounced phonemes and discuss them with each other until a

Table 2: The objective and subjective pronunciation accuracy comparisons in the Biaobei dataset.

Method	PER-O	PER-S	SER-S
Character	-	3.73%	30.50%
BERT Embedding [15]	-	4.03%	38.75%
NLR [16]	-	2.98%	26.50%
Phoneme (G2PM)	3.95%	1.39%	10.50%
Phoneme (pypinyin)	2.78%	1.14%	7.00%
Dict-TTS	2.12%	1.08%	6.50%
Dict-TTS (pre-trained)	**1.54%**	**0.79%**	**4.25%**

[5] `https://github.com/jik876/hifi-gan`

[6] We use *pypinyin* in the Biaobei dataset, *pyopenjtalk* in the JSUT dataset and *pycantonese* in the Common Voice (HK) dataset. More detailed information can be found in Appendix B.4



(a) GT (b) Character (c) BERT Embedding

(d) NLR (e) Phoneme (pypinyin) (f) Dict-TTS

Figure 5: Visualizations of the ground-truth and generated mel-spectrograms by different types of linguistic encoders. The corresponding text is "菱歌泛夜，嬉嬉钓叟莲娃", which means "The man picking water chestnut sings at night. The old man fishing and the girl picking lotus are laughing".

conclusion is reached. Note that for SER-S, the error rate is calculated in sentence level (e.g., a sentence with multiple errors will be counted only once). More details about these evaluations can be found in Appendix B.3. The results are shown in Table 1. It can be seen that Dict-TTS greatly surpasses the character-based baseline in three languages. Moreover, Dict-TTS achieves the comparable objective and subjective PER with the most popular grapheme-to-phoneme tools on two relatively larger datasets (Biaobei and Common Voice (HK)) and maintain a good pronunciation accuracy on a relatively small dataset (JSUT), which demonstrates the superiority of the explicit semantics matching in our S2PA module.

We also compare the pronunciation accuracy of our Dict-TTS with various types of systems, including: 1) a character-based system; 2) a BERT embedding based system [15], where the BERT derived embeddings are concatenated with the character embeddings; 3) NLR [16], a TTS system that directly injects BERT derived knowledge into the linguistic encoder without phoneme labels; 4) Phoneme (G2PM [38]), PortaSpeech with phoneme labels derived from G2PM (a powerful neural G2P system); 5) Phoneme (pypinyin), PortaSpeech with phoneme labels derived from pypinyin (one of the most popular Chinese G2P system). As shown in Table 2, Dict-TTS greatly surpasses the systems that implicitly model the semantic representations for character-to-pronunciation mapping like NLR [16] and shows comparable performance with phoneme-based systems. Since our Dict-TTS does not require any phoneme labels for training, we can pre-train Dict-TTS on a large-scaled ASR dataset [61] with a small amount of effort to improve its generalization capacity. It can be seen that the pronunciation accuracy of Dict-TTS on the Biaobei dataset is significantly improved by pre-training, which demonstrates the effectiveness of our unsupervised polyphone disambiguation framework.

4.3 Results of Audio Quality and Prosody

We compare the audio quality, audio prosody, and pitch accuracy[7] of Dict-TTS with the systems evaluated in Subsection 4.2. GT (the ground truth audio) and GT (voc.) (the ground truth audio that is firstly converted into mel-spectrogram and converted back to audio waveform using Hifi-GAN [29]) are also included in this experiment. We keep the text content consistent among different models to exclude other interference factors, only examining the audio quality or prosody. Each audio is listened by at least 20 testers. For audio quality and prosody, we conduct the mean opinion score (MOS) evaluation via Amazon Mechanical Turk.

[7]We compute the average dynamic time warping (DTW) [36] distances between the pitch contours of ground-truth speech and synthesized speech.

Table 4: Pronunciation accuracy, audio prosody and audio quality comparisons for ablation study.

Settings	PER-O	PER-S	SER-S	CMOS-P	CMOS-Q
Dict-TTS	2.12%	1.08%	6.50%	0.000	0.000
w/o Semantics	2.15%	1.13%	6.50%	-0.280	-0.135
w/o Gumbel-Softmax	-	1.19%	7.75%	-0.014	-0.021

We analyze the MOS in two aspects: MOS-P (Prosody: naturalness of pitch, energy, and duration) and MOS-Q (Quality: clarity, high-frequency, and original timbre reconstruction). We tell the tester to focus on one corresponding aspect and ignore the other aspect when scoring. We put more information about the subjective evaluation in Appendix B.3. As shown in Table 3, for audio quality, Dict-TTS significantly outperforms those TTS systems without phoneme labels and achieves a comparable performance with phoneme-based systems. And for audio prosody and pitch accuracy, Dict-TTS even surpasses the phoneme-based systems, which demonstrates the effectiveness of the extracted semantics from prior dictionary knowledge. We put more analyses on the naturalness of prosody in Appendix E.

Table 3: The audio performance (MOS-Q and MOS-P) and pitch accuracy comparisons. DTW denotes average dynamic time warping distances of pitch in ground-truth and synthesized audio. The mel-spectrograms are converted to waveforms using Hifi-GAN (V3) [29].

Method	MOS-P	MOS-Q	DTW
GT	4.48±0.03	4.40±0.04	-
GT (voc.)	4.37±0.04	4.26±0.04	-
Character	3.82±0.08	3.88±0.07	53.1
BERT Embedding [15]	3.88±0.07	3.63±0.10	55.0
NLR [16]	3.83±0.08	3.74±0.08	53.3
Phoneme (G2PM)	3.87±0.08	3.90±0.06	53.1
Phoneme (pypinyin)	3.89±0.08	**3.95±0.06**	52.6
Dict-TTS	**4.03±0.05**	3.91±0.04	**52.4**

We then visualize the mel-spectrograms generated by the above systems in Figure 5. We can see that Dict-TTS can generate mel-spectrograms with comparable details in harmonics, unvoiced frames, and high-frequency parts with the phoneme-based system, which results in similar natural sounds. Moreover, our Dict-TTS can capture more accurate local changes in pitch and speaking duration, indicating the effectiveness of introducing semantic representations in the dictionary.

4.4 Ablation Studies

We conduct ablation studies to demonstrate the effectiveness of designs in Dict-TTS, including the auxiliary semantic information and the Gumbel-Softmax sample strategy. We conduct pronunciation accuracy and CMOS (comparative mean opinion score) evaluations for these ablation studies. The results are shown in Table 4. We can see that CMOS-P drops when we remove the introduced semantic embeddings in Dict-TTS, indicating that the semantic information extracted from the dictionary can improve the audio prosody. Besides, to demonstrate the effectiveness of the Gumbel-Softmax sample strategy, we also compare the weighted sum of the pronunciation embeddings with the Gumbel-Softmax sample strategy. Since measuring PER-O requires one-hot vectors, we do not calculate the PER-O score for the weight-sum version of Dict-TTS. The results are shown in row 3 in Table 4. It can be seen that PER-S and SER-S increase when we use the weighted sum method. In the experiments, the weights of different pronunciations for some characters might be close to each other, which results in relatively worse performance in the subjective results. For example, the two pronunciations "ZH ANG3" and "CH ANG2" of the character "长" might be ambiguous when their weights are close to each other (e.g., 0.6 and 0.4). Therefore, to accurately model the subjective pronunciations, we utilize the Gumbel-Softmax function to sample the most likely pronunciation in both training and inference stages.

5 Conclusion

In this paper, we proposed Dict-TTS, an unsupervised framework for polyphone disambiguation in end-to-end text-to-speech systems. Dict-TTS uses a semantics-to-pronunciation attention (S2PA) module to explicitly extract the corresponding pronunciations and semantic information from prior dictionary knowledge. The S2PA module can be trained with the end-to-end TTS model with the

guidance of mel-spectrogram reconstruction loss without phoneme labels, which significantly reduces the cost of building a polyphone disambiguation system and further enables the efficient pre-training on large-scaled ASR datasets. Our experimental results in three languages show that Dict-TTS outperforms several strong G2P baseline models in terms of pronunciation accuracy and improves the prosody modeling of the baseline TTS system. Further comprehensive ablation studies demonstrate that our S2PA module successfully decomposes the character representation, pronunciation, and semantics for character-based TTS systems. However, the dictionary knowledge does not contain the tree-structured syntactic information of the input text sequence, which also affects the prosody modeling. Moreover, since we crawl the dictionary from online websites and do not make any specific changes, the performance of Dict-TTS can be further improved by a well-designed dictionary. In the future, we will try to inject syntactic information into Dict-TTS and extend it to more languages.

6 Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant No.62072397 and No.61836002), Zhejiang Natural Science Foundation (LR19F020006), Yiwise, and National Key R&D Program of China (Grant No.2020YFC0832505). We appreciate the support from Mindspore, which is a new deep learning computing framework. This work was also supported by Alibaba Group through Alibaba Innovative Research Program.

References

[1] Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4218–4222, 2020.

[2] Sercan Ö Arık, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, et al. Deep voice: Real-time neural text-to-speech. In *International Conference on Machine Learning*, pages 195–204. PMLR, 2017.

[3] Data Baker. Chinese standard mandarin speech corpus, 2017.

[4] Alan W Black, Kevin Lenzo, and Vincent Pagel. Issues in building general letter to sound rules. In *The third ESCA/COCOSDA workshop (ETRW) on speech synthesis*, 1998.

[5] Zexin Cai, Yaogen Yang, Chuxiong Zhang, Xiaoyi Qin, and Ming Li. Polyphone disambiguation for mandarin chinese using conditional neural network with multi-level embedding features. In Gernot Kubin and Zdravko Kacic, editors, *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 2110–2114. ISCA, 2019.

[6] Moon-Jung Chae, Kyubyong Park, Jinhyun Bang, Soobin Suh, Jonghyuk Park, Namju Kim, and Longhun Park. Convolutional sequence to sequence model with non-sequential greedy decoding for grapheme to phoneme conversion. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2486–2490. IEEE, 2018.

[7] Jiawei Chen, Xu Tan, Jian Luan, Tao Qin, and Tie-Yan Liu. Hifisinger: Towards high-fidelity neural singing voice synthesis. *CoRR*, abs/2009.01776, 2020.

[8] Yi-Chang Chen, Yu-Chuan Chang, Yen-Cheng Chang, and Yi-Ren Yeh. g2pw: A conditional weighted softmax bert for polyphone disambiguation in mandarin. *arXiv preprint arXiv:2203.10430*, 2022.

[9] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, 2019.

[10] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, 2020.

[11] Dongyang Dai, Zhiyong Wu, Shiyin Kang, Xixin Wu, Jia Jia, Dan Su, Dong Yu, and Helen Meng. Disambiguation of chinese polyphones in an end-to-end framework with semantic features extracted by pre-trained bert. In *Interspeech*, pages 2090–2094, 2019.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[13] Jeff Donahue, Sander Dieleman, Mikolaj Binkowski, Erich Elsen, and Karen Simonyan. End-to-end adversarial text-to-speech. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[14] Isaac Elias, Heiga Zen, Jonathan Shen, Yu Zhang, Ye Jia, R. J. Skerry-Ryan, and Yonghui Wu. Parallel tacotron 2: A non-autoregressive neural TTS model with differentiable duration modeling. In Hynek Hermansky, Honza Cernocký, Lukás Burget, Lori Lamel, Odette Scharenborg, and Petr Motlícek, editors, *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pages 141–145. ISCA, 2021.

[15] Tomoki Hayashi, Shinji Watanabe, Tomoki Toda, Kazuya Takeda, Shubham Toshniwal, and Karen Livescu. Pre-trained text embeddings for enhanced text-to-speech synthesis. In *INTERSPEECH*, pages 4430–4434, 2019.

[16] Mutian He, Jingzhou Yang, Lei He, and Frank K Soong. Neural lexicon reader: Reduce pronunciation errors in end-to-end tts by leveraging external textual knowledge. *arXiv preprint arXiv:2110.09698*, 2021.

[17] John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, 2019.

[18] Rem Hida, Masaki Hamada, Chie Kamada, Emiru Tsunoo, Toshiyuki Sekiya, and Toshiyuki Kumakura. Polyphone disambiguation and accent prediction using pre-trained language models in japanese tts front-end. *arXiv preprint arXiv:2201.09427*, 2022.

[19] Feng-Long Huang. Disambiguating effectively chinese polyphonic ambiguity based on unify approach. In *2008 International Conference on Machine Learning and Cybernetics*, volume 6, pages 3242–3246. IEEE, 2008.

[20] Rongjie Huang, Max WY Lam, Jun Wang, Dan Su, Dong Yu, Yi Ren, and Zhou Zhao. Fastdiff: A fast conditional diffusion model for high-quality speech synthesis. *arXiv preprint arXiv:2204.09934*, 2022.

[21] Rongjie Huang, Yi Ren, Jinglin Liu, Chenye Cui, and Zhou Zhao. Generspeech: Towards style transfer for generalizable out-of-domain text-to-speech synthesis. *arXiv preprint arXiv:2205.07211*, 2022.

[22] Rongjie Huang, Zhou Zhao, Huadai Liu, Jinglin Liu, Chenye Cui, and Yi Ren. Prodiff: Progressive fast diffusion model for high-quality text-to-speech. *arXiv preprint arXiv:2207.06389*, 2022.

[23] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[24] John T Jensen. *Principles of generative phonology: An introduction*, volume 250. John Benjamins Publishing, 2004.

[25] Ye Jia, Heiga Zen, Jonathan Shen, Yu Zhang, and Yonghui Wu. Png BERT: augmented BERT on phonemes and graphemes for neural TTS. In Hynek Hermansky, Honza Cernocký, Lukás Burget, Lori Lamel, Odette Scharenborg, and Petr Motlícek, editors, *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pages 151–155. ISCA, 2021.

[26] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2019.

[27] Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon. Glow-tts: A generative flow for text-to-speech via monotonic alignment search. *Advances in Neural Information Processing Systems*, 33:8067–8077, 2020.

[28] Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, pages 5530–5540. PMLR, 2021.

[29] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33:17022–17033, 2020.

[30] Younggun Lee, Suwon Shon, and Taesu Kim. Learning pronunciation from a foreign language in speech synthesis networks. *arXiv preprint arXiv:1811.09364*, 2018.

[31] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. Neural speech synthesis with transformer network. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6706–6713. AAAI Press, 2019.

[32] Jinglin Liu, Chengxi Li, Yi Ren, Feiyang Chen, and Zhou Zhao. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11020–11028, 2022.

[33] Jinke Liu, Weiguang Qu, Xuri Tang, Yizhe Zhang, and Yuxia Sun. Polyphonic word disambiguation with machine learning approaches. In *2010 Fourth International Conference on Genetic and Evolutionary Computing*, pages 244–247. IEEE, 2010.

[34] Wei Liu, Xiyan Fu, Yue Zhang, and Wenming Xiao. Lexicon enhanced chinese sequence labeling using bert adapter. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5847–5858, 2021.

[35] Maryanne Martin, Gregory V Jones, Douglas L Nelson, and Louise Nelson. Heteronyms and polyphones: Categories of words with multiple phonemic representations. *Behavior Research Methods & Instrumentation*, 13(3):299–307, 1981.

[36] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[37] Jongseok Park, Kyubyong & Kim. g2pe. `https://github.com/Kyubyong/g2p`, 2019.

[38] Kyubyong Park and Seanie Lee. g2pm: A neural grapheme-to-phoneme conversion package for mandarin chinese based on a new open benchmark dataset. *arXiv preprint arXiv:2004.03136*, 2020.

[39] Matthew E Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, 2018.

[40] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[41] Yi Ren, Jinglin Liu, and Zhou Zhao. Portaspeech: Portable and high-quality generative text-to-speech. *Advances in Neural Information Processing Systems*, 34, 2021.

[42] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech: Fast, robust and controllable text to speech. *Advances in Neural Information Processing Systems*, 32, 2019.

[43] Changhao Shan, Lei Xie, and Kaisheng Yao. A bi-directional lstm approach for polyphone disambiguation in mandarin chinese. In *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 1–5. IEEE, 2016.

[44] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

[45] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[46] Alex Sokolov, Tracy Rohlin, and Ariya Rastrow. Neural machine translation for multilingual grapheme-to-phoneme conversion. *arXiv preprint arXiv:2006.14194*, 2020.

[47] Ryosuke Sonobe, Shinnosuke Takamichi, and Hiroshi Saruwatari. Jsut corpus: free large-scale japanese speech corpus for end-to-end speech synthesis. *arXiv preprint arXiv:1711.00354*, 2017.

[48] Hao Sun, Xu Tan, Jun-Wei Gan, Hongzhi Liu, Sheng Zhao, Tao Qin, and Tie-Yan Liu. Token-level ensemble distillation for grapheme-to-phoneme conversion. *arXiv preprint arXiv:1904.03446*, 2019.

[49] Hao Sun, Xu Tan, Jun-Wei Gan, Sheng Zhao, Dongxu Han, Hongzhi Liu, Tao Qin, and Tie-Yan Liu. Knowledge distillation from bert in pre-training and fine-tuning for polyphone disambiguation. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 168–175. IEEE, 2019.

[50] Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. A survey on neural speech synthesis. *arXiv preprint arXiv:2106.15561*, 2021.

[51] Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, 2019.

[52] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *The 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13-15 September 2016*, page 125. ISCA, 2016.

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[54] Yonghe Wang, Feilong Bao, Hui Zhang, and Guanglai Gao. Joint alignment learning-attention based model for grapheme-to-phoneme conversion. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7788–7792. IEEE, 2021.

[55] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. Tacotron: Towards end-to-end speech synthesis. *Proc. Interspeech 2017*, pages 4006–4010, 2017.

[56] Tingyu Xia, Yue Wang, Yuan Tian, and Yi Chang. Using prior knowledge to guide bert's attention in semantic textual matching tasks. In *Proceedings of the Web Conference 2021*, pages 2466–2475, 2021.

[57] Bing Yang, Jiaqi Zhong, and Shan Liu. Pre-trained text representations for improving front-end text processing in mandarin text-to-speech synthesis. In *INTERSPEECH*, pages 4480–4484, 2019.

[58] Kaisheng Yao and Geoffrey Zweig. Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. In *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, pages 3330–3334. ISCA, 2015.

[59] Zhenhui Ye, Zhou Zhao, Yi Ren, and Fei Wu. SyntaSpeech: Syntax-Aware Generative Adversarial Text-to-Speech. *arXiv e-prints*, page arXiv:2204.11792, April 2022.

[60] Mingzhi Yu, Hieu Duy Nguyen, Alex Sokolov, Jack Lepird, Kanthashree Mysore Sathyendra, Samridhi Choudhary, Athanasios Mouchtaris, and Siegfried Kunzmann. Multilingual grapheme-to-phoneme conversion with byte representation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8234–8238. IEEE, 2020.

[61] Binbin Zhang, Hang Lv, Pengcheng Guo, Qijie Shao, Chao Yang, Lei Xie, Xin Xu, Hui Bu, Xiaoyu Chen, Chenchen Zeng, et al. Wenetspeech: A 10000+ hours multi-domain mandarin corpus for speech recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6182–6186. IEEE, 2022.

[62] Guangyan Zhang, Kaitao Song, Xu Tan, Daxin Tan, Yuzi Yan, Yanqing Liu, Gang Wang, Wei Zhou, Tao Qin, Tan Lee, et al. Mixed-phoneme bert: Improving bert with mixed phoneme and sup-phoneme representations for text to speech. *arXiv preprint arXiv:2203.17190*, 2022.

[63] Haiteng Zhang, Huashan Pan, and Xiulin Li. A mask-based model for mandarin chinese polyphone disambiguation. In *INTERSPEECH*, pages 1728–1732, 2020.

[64] Hong Zhang, JiangSheng Yu, WeiDong Zhan, and ShiWen Yu. Disambiguation of chinese polyphonic characters. In *The First International Workshop on MultiMedia Annotation (MMA2001)*, volume 1, pages 30–1, 2001.

[65] Yang Zhang, Liqun Deng, and Yasheng Wang. Unified mandarin tts front-end based on distilled bert model. *arXiv preprint arXiv:2012.15404*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix H in the supplementary materials.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplementary materials.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Subsection 4.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report confidence intervals of subjective metric results and describe the random seed settings in Appendix in the supplementary materials.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Subsection 4.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Subsection 4.1.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [Yes] See Appendix B.3 in the supplementary materials.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [Yes] See Appendix B.3 in the supplementary materials.

Appendices

A Details of Models

In this section, we describe details about the dictionary construction, the architecture of the semantic encoder and linguistic encoder, and multi-length discriminator. For further information about the variational generator, one could refer to PortaSpeech [41]. We also describe our modifications to model architecture to support the pre-training task of Dict-TTS with large-scaled ASR datasets.

A.1 Dictionary Construction

The Chinese dictionary used in the experiments is obtained from `https://github.com/yihui/zdict`, which is crawled from `https://www.zdic.net/zd/zb/ty/`. The Japanese dictionary is crawled from `https://dictionary.goo.ne.jp` and the Cantonese dictionary is crawled from `https://humanum.arts.cuhk.edu.hk/Lexis/lexi-can`. The crawled dictionaries are processed into the format described in SubSection 3.3.

We use a pre-trained cross-lingual language model [10] to extract the semantic context information k of every gloss entry e and store them in the disk for computational efficiency[8]. Note that our Dict-TTS does not need the heavy BERT model in the inference stage. As is pointed out by the previous works, lower layers of BERT are found to perform broad attention across all pairs of words or encode local syntax [9, 39] and middle layers are found to mostly capture transferable syntactic and semantic knowledge [17, 51]. But the upper layers are specifically tuned towards the pre-training tasks of BERT [56]. Thus, we use the average of the first-layer word embeddings and the later 8-layer contextual BERT representations as the extracted knowledge.

A.2 Architecture of the Semantic Encoder and Linguistic Encoder

As shown in Figure 6, the semantic encoder and linguistic encoder of our Dict-TTS are both stacks of feed-forward Transformer layers with relative position encoding [44] following PortaSpeech [41]. And there are four layers of the semantic encoder and four layers of the linguistic encoder in our Dict-TTS.

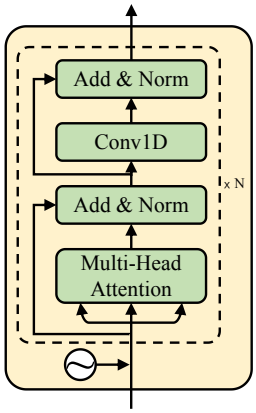


Figure 6: The detailed architecture of the semantic encoder and linguistic encoder.

A.3 Modifications for Pre-training Task

The vanilla architecture of PortaSpeech is tested on LJSpeech (a single-speaker dataset). By introducing a group of learnable speaker embeddings to represent the speakers' timbre, pronunciation habits, and other features, it can be trained on multi-speaker datasets. However, since large-scaled ASR datasets may not have explicit speaker information, the following modifications should be made: we extract the speaker embeddings from audio samples using resemblyzer[9] and feed them into the variational generator and duration predictor. We pre-train our DictTTS on the WenetSpeech dataset [61], which is a multi-domain corpus for speech recognition. It takes 600k steps for pre-training until convergence. As is shown in Table 2, the semantic comprehension and the generalization capacity of our Dict-TTS are significantly improved by our pre-training process.

B Detailed Experimental Settings

In this section, we describe more model configurations and details in subjective evaluation.

[8]The model used in our experiments can be downloaded from `https://huggingface.co/xlm-roberta-base`. One could use monolingual BERT to enhance the performance.

[9]`https://github.com/resemble-ai/Resemblyzer`

B.1 Dict-TTS Model Configurations

We list the model hyper-parameters of Dict-TTS in Table 5.

Table 5: Hyperparameters of Dict-TTS models.

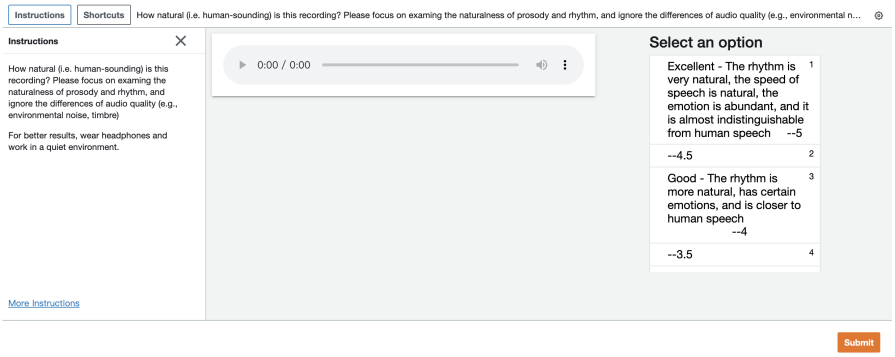
Hyper-parameter		Dict-TTS	Number of parameters
Semantic Encoder	Character Embedding	192	4.907M
	Semantic Encoder Layers	4	
	Hidden Size	192	
	Conv1D Kernel	5	
	Conv1D Filter Size	768	
S2PA Module	Hidden Size	192	0.404M
Linguistic Encoder	Linguistic Encoder Layers	4	3.371M
	Hidden Size	192	
	Conv1D Kernel	5	
	Conv1D Filter Size	768	
Variational Generator	Encoder Layers	8	7.516M
	Decoder Layers	4	
	Encoder/Decoder Kernel	5	
	Encoder/Decoder channel size	192	
	Latent Size	16	
	Prior Flow Layers	4	
	Prior Flow Conv1D Kernel	3	
	Prior Flow Conv1D Channel Size	64	
Multi-Length Discriminator	Number of Discriminators	3	0.927M
	Window Size	32, 64, 128	
	Conv2D Layers	3	
	Hidden Size	192	
Total Number of Parameters			17.125M

B.2 Baseline Model Configurations

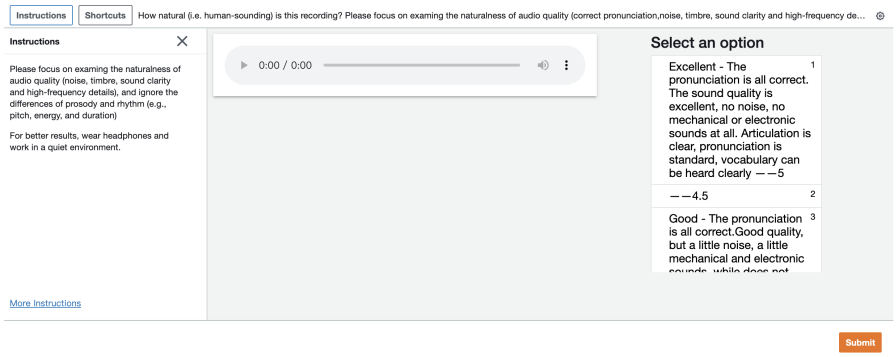
The baseline systems in our experiments can be divided into: 1) phoneme-based systems; 2) character-based systems. For phoneme-based systems, we use the mixture alignment proposed in PortaSpeech [41], which takes phoneme sequence as inputs but utilize both soft phoneme-level and hard word-level duration for mixture alignment. We use a 4-layer phoneme encoder and a 4-layer word encoder following the implementation of PortaSpeech [41]. For character-based systems, we use an 8-layer character encoder for fair comparisons. The encoders above are stacks of feed-forward Transformer layers with relative position encoding [44] following PortaSpeech [41]. Other parts of the architecture are kept the same as Dict-TTS to exclude other interference factors.

B.3 Details in Subjective Evaluation

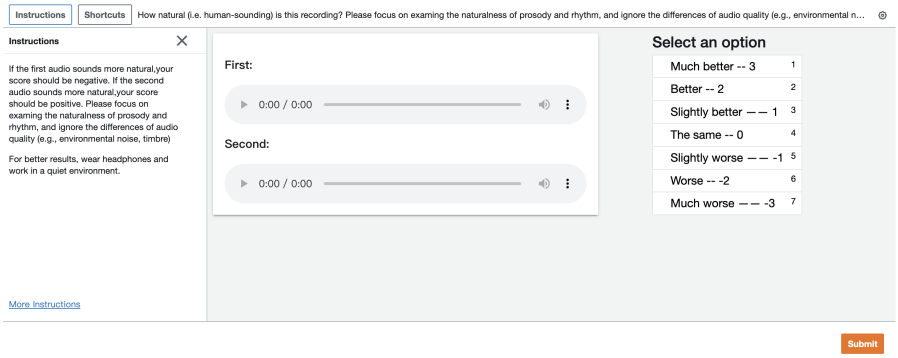
Audio Quality and Prosody We perform the audio quality and prosody evaluation on Amazon Mechanical Turk (MTurk). For each dataset, we randomly select 50 texts from the test set and use the TTS systems to generate the audio samples. Each audio has been listened to by at least 20 listeners. For MOS, each tester is asked to evaluate the subjective naturalness of a sentence on a 1-5 Likert scale. For CMOS, listeners are asked to compare pairs of audio generated by systems A and B, indicate which of the two audio they prefer, and choose one of the following scores: 0 indicating no difference, 1 indicating a slight difference, 2 indicating a significant difference and 3 indicating a very large difference. For audio quality evaluation (MOS-Q and CMOS-Q), we tell listeners to "*focus on examining the naturalness of audio quality (e.g., noise, timbre, sound clarity, and high-frequency details), and ignore the differences of prosody and rhythm (e.g., pitch, energy, and duration)*". For prosody evaluations (MOS-P and CMOS-P), we tell listeners to "*focus on examining the naturalness of prosody and rhythm (e.g., pitch, energy, and duration), and ignore the differences in audio quality (e.g., noise, timbre, sound clarity, and high-frequency details)*". The screenshots of instructions for



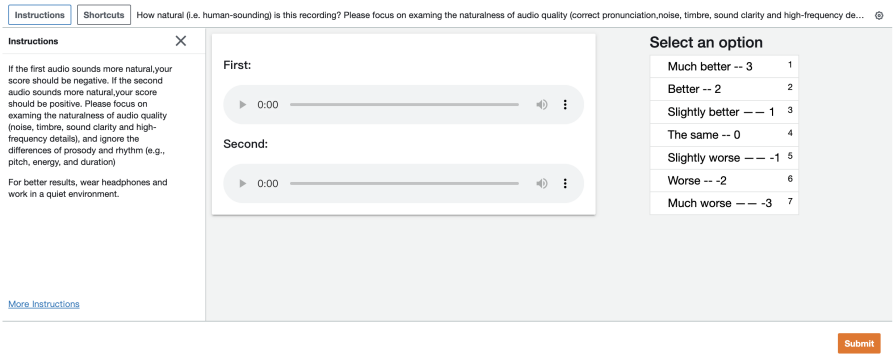
(a) Screenshot of MOS-P testing.



(b) Screenshot of MOS-Q testing.



(c) Screenshot of CMOS-P testing.



(d) Screenshot of CMOS-Q testing.

Figure 7: Screenshots of audio quality and prosody evaluations.

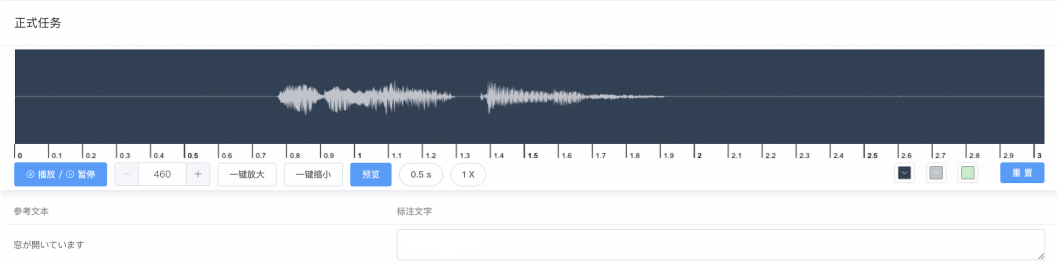


Figure 8: Screenshot of pronunciation accuracy evaluation.

testers are shown in Figure 7. We paid $8 to participants hourly and totally spent about $500 on participant compensation.

Pronunciation Accuracy We perform the pronunciation accuracy evaluation on MolarData[10]. For each dataset, we use all texts in the test set to generate the audio samples. Each audio has been listened to by at least 4 language experts. Each tester is asked to carefully listen to the audio multiple times, write down the mispronounced phonemes, and discuss with each other until a conclusion is reached. The screenshots of instructions for testers are shown in Figure 8. We paid $15 to participants hourly and totally spent about $800 on participant compensation. A small subset of speech samples used in the test is available at `https://dicttts.github.io/DictTTS-Demo/`.

B.4 Details of the G2P tools used in the experiments

We use *pypinyin (0.46.0)*[11] and G2PM[12] in Biaobei dataset. Until 20 May, 2022, the latest version of pypinyin is 0.46.0. And it is worth noting that the pronunciation accuracy is greatly improved when we update the version from 0.36.0 to 0.46.0. In JSUT dataset, we use *pyopenjtalk (0.2.0)*[13]. And in Common Voice (HK) dataset, we use *pycantonese (3.4.0)*[14].

B.5 Details of Datasets

In this subsection, we mainly describe the distribution characteristics of polyphones in the datasets used in our experiments, including: 1) Biaobei [3]; 2) JSUT [47] and 3) Common Voice (HK) [1]. As shown in Figure 9, we calculate the number of polyphones in every sentence. The average sentence lengths of the Biaobei, JSUT, and Common Voice (HK) datasets are 18, 29, and 10. Among these datasets, the most frequent number of polyphones appearing in a sentence is three to five and some sentences even include eight (or more) polyphones, indicating the importance of polyphone disambiguation.

B.6 Error Bars and Random Seeds

For the experiments of the audio quality and prosody, we report confidence intervals of subjective metric results in Table 3. For the experiments of the pronunciation accuracy, we ran the experiments 10 times with 10 different random seeds ($[1234, 1111, 2222, 3333, 4444, 5555, 6666, 7777, 8888, 9999]$) and obtained the averaged results.

C Visualization of Attention Weights

We put some semantics attention visualizations in Figure 10. We can see that Dict-TTS can create reasonable text-to-dictionary alignments, which improves the performance of polyphone disambiguation and helps the semantics comprehension for end-to-end TTS systems.

[10] `https://www.molardata.com/en.html`
[11] `https://github.com/mozillazg/python-pinyin`
[12] `https://github.com/kakaobrain/g2pM`
[13] `https://github.com/r9y9/pyopenjtalk`
[14] `https://github.com/jacksonllee/pycantonese`

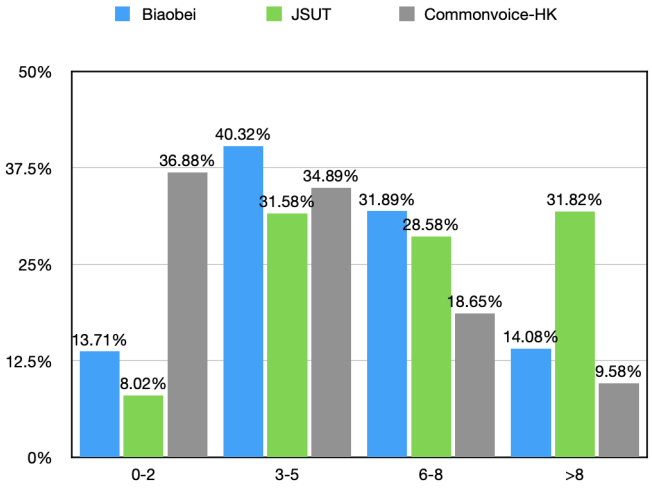
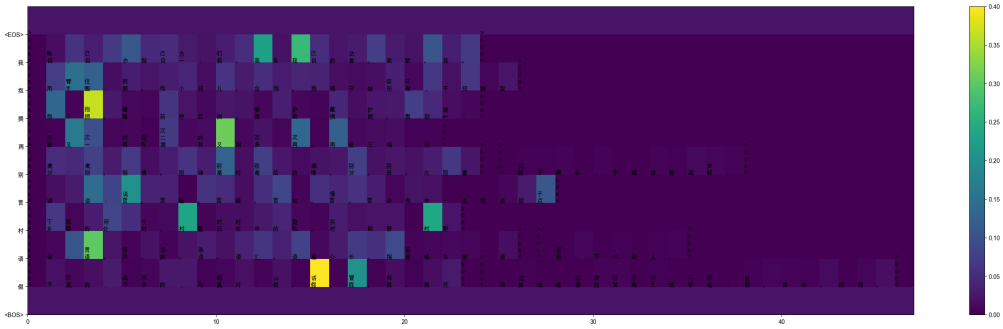
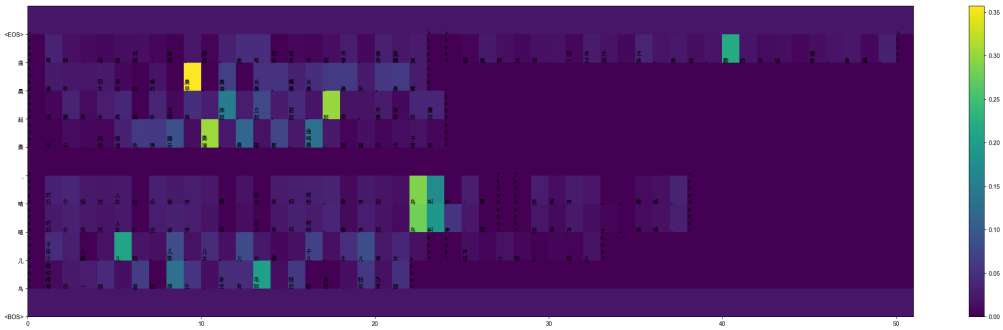


Figure 9: The illustration of the number of polyphone in a sentence. The horizontal axis is the frequency of polyphones in a sentence and the vertical axis is the number of sentences shown by percentage.



(a) Biaobei-000001



(b) Biaobei-000002

Figure 10: Visualizations of the semantics attention weights in Biaobei dataset.

D The Definition of Polyphone and Heteronym

Polyphones are letters or characters having more than one phonetic value, while heteronyms are words that have two (or more) different possible pronunciations that are associated with two (or more) different meanings [35]. According to the previous research [63], the difficulty of polyphone disambiguation in logographic languages mainly lies in heteronyms. Thus, the previous works [11, 49, 38] usually use "polyphone disambiguation" to refer to "heteronym disambiguation" for those languages. In this work, we use "polyphone disambiguation" to refer to "heteronym disambiguation" following the previous works.

E Analyses on Prosodic Realization

The DTW measure would be impacted by both lexical tone and prosodic realization. In order to further analyse the naturalness of prosody and rhythm for Dict-TTS and baseline systems, we evaluate the duration errors and character-level average pitch errors. For duration errors, we calculate the MSE of character-level durations. For character-level average pitch errors, we firstly calculate the mean pitch for each character's region in the mel spectrogram according to the Montreal Forced Aligner (MFA) to remove the influence of lexical tone, and then we calculate the MSE of the mean pitch sequences. The results on the Biaobei dataset are shown in Table 6. It can be seen that the duration error and the average pitch error of Dict-TTS are significantly lower than those of the baseline systems, demonstrating the effectiveness of the extracted semantics from prior dictionary knowledge.

Table 6: Duration error and average pitch error comparisons on the Biaobei dataset.

Method	Duration Error (ms)	Average Pitch Error
Character	36.2	1424.6
BERT Embedding	35.7	1312.1
NLR	36.4	1414.3
Phoneme (G2PM)	35.8	1341.7
Phoneme (pypinyin)	35.3	1308.8
Dict-TTS	**34.4**	**1232.3**

F Adding Rules to the Pronunciation Weights

There are some pronunciation rules (like "sandhi rules") that can not be learned from the dictionary. For example, in Mandarin, "一" before tone4 should be "Y I2" (e.g., "一段") and when "一" is an ordinal word, it should be "Y I1" (e.g., "一四九五年"). According to these pronunciation rules, we can obtain the correct pronunciation labels for some specific characters based on the part-of-speech (POS) tags of the input character sequence. After we obtain the correct pronunciation labels for these specific characters, we can directly force the pronunciation weights of these characters to be the ground truth values.

G Polyphone Disambiguation for Various Languages

The polyphone disambiguation problem is critical in logographic languages such as Chinese, but is less problematic in phonograms like English.

For logographic languages like Chinese, although the lexicon can cover nearly all the characters, there are a lot of polyphones that can only be decided according to the context of a character. Thus, G2P conversion in this kind of languages is mainly responsible for polyphone disambiguation, which decides the appropriate pronunciation based on the current word context. Therefore, polyphone disambiguation is crucial in these languages and our method is an effective solution for the polyphone disambiguation problem in these languages. For alphabetic languages like English, lexicon cannot cover the pronunciations of all the words. Thus, the G2P conversion for English is mainly responsible for generating the pronunciations of out-of-vocabulary words [50]. Although the polyphone disambiguation is less problematic in these languages, our methods can still be used as the modules

to retrieve the correct pronunciation for polyphones and heteronyms in their G2P process (e.g., the Algorithm step 2 in `https://github.com/Kyubyong/g2p`).

In our experiments, JSUT dataset is a mixture of phonograms and logograms, which is different from Biaobei and Common Voice (HK). Japanese writing system consists of two types of characters: the kanji (漢字) and the syllabic kana – hiragana (平仮名) and katakana (片仮名). In our analysis, 32.42% of the characters in JSUT dataset are kanji. The pronunciations of a part of the kanji can not only be specified by the semantic information and should be specified by empirical pronunciation rules. For example, most kanji (漢字) can be pronounced multiple ways: on-yomi (音読み) and kun-yomi (訓読み). Although the compound kanji usually uses on-yomi and one kanji probably uses kunyomi, the different readings are largely just chosen empirically in practice. Our Dict-TTS has the potential to work only for the kanji whose pronunciation should be specified based on the semantic meaning. Due to the characteristics of Japanese writing systems, in Table 1, although Dict-TTS surpasses the character-based system, it does not show comparable performance with the open source G2P module in Japanese. But as shown in Section 3.4, our method is compatible with the predefined rules from language experts by directly adding specific rules to pronunciation weight. We are sure that the performance of our method can be further improved by introducing the pronunciation rules in Japanese (e.g., the rules in the rule-based G2P baseline "pyopenjtalk").

H Potential Negative Societal Impacts

Dict-TTS improves the pronunciation accuracy and prosody of the synthesized speech voice and lowers the requirements for G2P conversion, which may cause unemployment for people with related occupations. Besides, the production of fake speeches may cause voice security issues. Further efforts in automatic speaker verification should be made to improve voice security.