



# Neural Lexicon Reader: Reduce Pronunciation Errors in End-to-end TTS by Leveraging External Textual Knowledge

Mutian He<sup>1\*</sup> Jingzhou Yang<sup>2</sup> Lei He<sup>2</sup> Frank K. Soong<sup>2</sup>

<sup>1</sup> The Hong Kong University of Science and Technology <sup>2</sup> Microsoft China

mhear@connect.ust.hk, {jingy, helei, frankkps}@microsoft.com

## Abstract

End-to-end TTS requires a large amount of speech/text paired data to cover all necessary knowledge, particularly how to pronounce different words in diverse contexts, so that a neural model may learn such knowledge accordingly. But in real applications, such high demand of training data is hard to be satisfied and additional knowledge often needs to be injected manually. For example, to capture pronunciation knowledge on languages without regular orthography, a complicated grapheme-to-phoneme pipeline needs to be built based on a large structured pronunciation lexicon, leading to extra, sometimes high, costs to extend neural TTS to such languages. In this paper, we propose a framework to learn to automatically extract knowledge from unstructured external resources using a novel Token2Knowledge attention module. The framework is applied to build a TTS model named Neural Lexicon Reader that extracts pronunciations from raw lexicon texts in an end-to-end manner. Experiments show the proposed model significantly reduces pronunciation errors in low-resource, end-to-end Chinese TTS, and the lexicon-reading capability can be transferred to other languages with a smaller amount of data.

**Index Terms:** speech synthesis, neural net architecture

## 1. Introduction

End-to-end neural models have greatly pushed the frontiers of the area of text-to-speech (TTS) [1]. However, these models still suffer from high data demands. As a machine learning method, a large amount of data is generally required to cover all knowledge necessary to produce speech: pronunciation of each grapheme under different contexts, semantic meanings of words to produce natural prosody, etc. The cost to collect such a complete dataset can be unbearable, and knowledge learned from limited data may fail to extrapolate to all cases, not to say that to internalize (i.e. learn) all these knowledges can be challenging for a single neural model.

Acquisition of pronunciation knowledge is a noticeable aspect, and its failure could be the root of TTS errors in end-to-end models like reading “chaos” as “CH-AW-S” or “San Jose” as “S-AA-N JH-OW-S”. It is because not only the data may be unable to cover these edge cases, but also the model may fail to learn such outliers sparsely distributed in the data. These two might be solved by a simple lexicon look-up, but how about “desert” in “get their just desert” versus “lost in a desert”? It can be even more severe: Thousands of Chinese characters are used in Japanese, each has multiple readings depending on the context. Chinese uses more characters that take native speakers years to memorize. Arabic and Hebrew scripts typically omit vowels, leaving the reader to decide the meaning and pronunciation. On these languages without phonemic orthogra-

\* Work done at Microsoft. Corresponding author: Lei He.

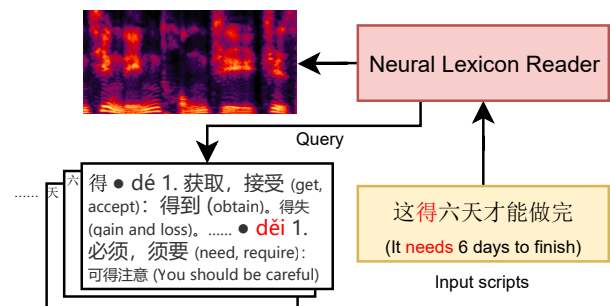


Figure 1: *Neural Lexicon Reader under the framework, in which raw lexicon text of each character is used as external knowledge. The lexicon contains readings as well as meanings and usages for each reading. The character “得” takes the second reading dēi here, since it means “need” in the sentence. English translations are indicated.*

phy, fully end-to-end TTS models are often less reliable, needing an extra manually-built complicated grapheme-to-phoneme (G2P) pipeline based on a combination of hand-crafted rules, structured G2P-oriented lexicons, and machine learning models [2, 3, 4]. In this way, additional knowledge and labor from language experts are introduced, which take considerable costs on every single language, unscalable to be extended to countless languages in the real world.

Following research on scalability-centered low-resource TTS [5], we aim at scaling end-to-end TTS to various languages using minimal per-language human efforts. Given the difficulty both to directly internalize all the knowledge from data and to leverage external knowledge through language expertise, we consider it desirable if external knowledge could be **automatically** extracted and utilized. This is inspired by machine reading comprehension (MRC) and question answering (QA): With rich knowledge learned, BERT may answer questions like “The capital city of Italy is [MASK]” by themselves [6]. But the answers are often imprecise as it is difficult and inefficient for a neural network to memorize countless definite or “hard” answers (like the capital of Italy) sporadically appeared in training data. Instead, open-domain QA models are more reliable on this task. They do not store such knowledge inside themselves, but develop a “soft” ability to read and understand related external texts (like Wikipedia articles), and then find the correct answer to the question from the article [7, 8].

TTS resembles QA in that both the hard knowledge (e.g. pronunciations and semantics of each word) and soft ability (e.g. to choose the exact reading) are required. Therefore, we design a general framework with novel Token2Knowledge attention to automatically extract relevant knowledge from external texts. To exemplify, we propose Neural Lexicon Reader

(NLR) shown in Figure 1, which extracts pronunciations of words in the input script from raw lexicon texts without a G2P pipeline using structured rules hand-crafted on the language. It mimics humans that when a child doesn’t know how to read “chaos”, the child will look it up in a dictionary. More, general-purpose dictionaries are available in most languages, hence the framework can be easily extended to new languages.

Moreover, it is more flexible, controllable, and interpretable: The model is targeted to develop a more general lexicon-reading capability to be used in runtime that is less dependent on the exact content of scripts and lexicons seen during training. Hence even for a trained model, pronunciations can be manipulated and new words can be actively added by updating the lexicon. Empirical studies on Mandarin Chinese show that NLR produces better intelligibility than character-based models, particularly with limited data. It also helps to resolve heteronyms or polyphones, and generalizes to rare and even unseen characters well, which are effectively zero-shot learning. Furthermore, the lexicon-reading capability is less language-specific and can be better transferred to related languages like Cantonese and Japanese with much less data.

To conclude, our contributions are three-fold:

1. We propose a general framework to leverage external knowledge in neural TTS using Token2Knowledge attention;
2. We apply the framework to achieve fully end-to-end TTS on non-phonemic scripts without the need of particular language expertise for building G2P;
3. We demonstrate its general lexicon-reading ability using low-resource and cross-language transfer scenarios.

Code, audio samples, and the model using open data are available at [mutiann.github.io/papers/nlr](https://mutiann.github.io/papers/nlr).

## 2. Methods

We follow [5] to use a transformer model, but with character inputs and 4 decoder layers, and Token2Knowledge attention introduced in the encoder, as illustrated in Figure 2: For each token  $c_j$ , possibly a character, its relevant textual knowledge  $T_{j,1}, T_{j,2}, \dots, T_{j,m}$  (possibly the lexicon entry of the word it belongs to) is passed through a **knowledge encoder** to create two sequences of hidden representations: **keys**  $k_{j,1}, k_{j,2}, \dots, k_{j,m}$  and **values**  $v_{j,1}, v_{j,2}, \dots, v_{j,m}$ . Then, for the hidden state  $h_{i,j}$  at encoder layer  $i$ , we compute a context vector of

$$c_{i,j} = \text{attention}(h_{i,j}, [k_{j,1}, \dots, k_{j,m}], [v_{j,1}, \dots, v_{j,m}]) \quad (1)$$

by the standard multi-head attention layer that extracts relevant values according to the matchness between  $h_{i,j}$  and the key, followed by  $h'_{i,j} = \text{Layernorm}(c_{i,j} + h_{i,j})$ . This is similar to the transformer decoder, in which the decoder states perform not only self-attention but also attention with encoder states. The difference is that, in the decoder, all elements in the sequence of decoder states attend to the same sequence of encoder states. While in Token2Knowledge Attention, different elements attend to different encoded external knowledge.

The framework is flexible: the form of knowledge, the knowledge encoder, and the way to select relevant knowledge can be decided according to the scenario. As for NLR to extract pronunciation knowledge in our experiments, raw texts from the corresponding lexicon entry are used as knowledge, and the pre-trained multilingual language model XLM-R [9] is used as the knowledge encoder. Final-layer embeddings from XLM-R are used as **keys** to capture the semantics of the lexicon texts, so as to match the semantics between input texts and lexicon texts.

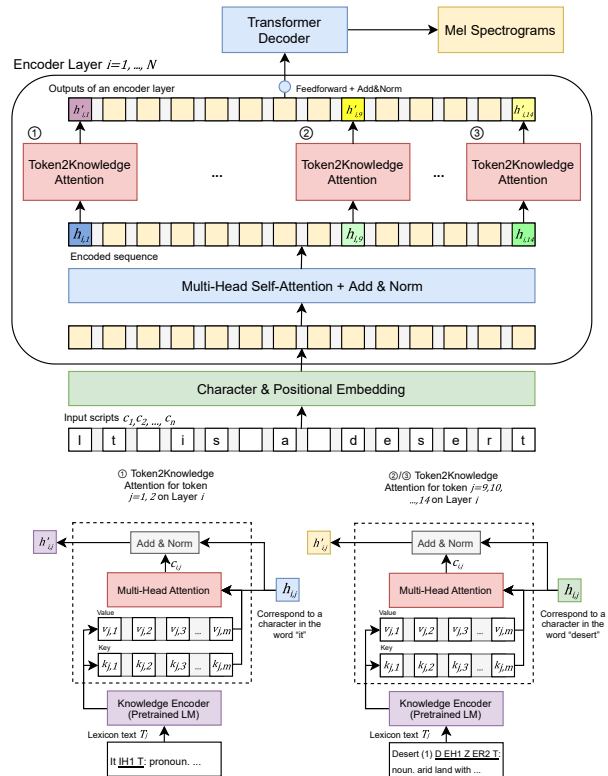


Figure 2: Model architecture, using an English sentence and a dictionary with ARPABET pronunciation notation as an example. ① is computed upon an element in the encoded sequence that corresponds to “I” in “It”, while ② corresponds to “d” and ③ to “t” in “desert”. Therefore ② and ③ use the same dictionary entry for “desert” for the Token2Knowledge Attention. The Token2Knowledge Attention modules are identical using shared parameters, but lexicon texts and input elements from the encoded sequence are different.

While to represent pronunciations, exact identities of characters (such as those in the phonemic notations) in the lexicon are valuable to present in attention outputs. Therefore, **values** are built by a concatenation of final-layer and first-layer embeddings of XLM-R, covering both deep (semantic) and shallow (identity) information. For unknown tokens and punctuation without lexicon entries, keys and values are both zero vectors. Token2Knowledge layers are used in all except the first encoder layers. Lexicon texts are from online dictionary websites.<sup>1</sup>

## 3. Experiments

We experimented on internal data of multiple East Asian languages, which exemplify the challenges we try to handle, as capturing their pronunciation is difficult, particularly in low-resource cases: There are no general rules, requiring case-by-case memorization; Readings are context-dependent, so a simple lexicon-lookup is not enough. They are also widely spoken but rather diverse, many of them (like various Sinitic dialects) under-resourced, making the scenario realistic. Nevertheless, our method is applicable beyond them, as many other languages

<sup>1</sup> [dictionary.goo.ne.jp](https://dictionary.goo.ne.jp) for Japanese, [humanum.arts.cuhk.edu.hk/Lexis/lexi-can](https://humanum.arts.cuhk.edu.hk/Lexis/lexi-can) for Cantonese, and [zdic.net](https://zdic.net) for Mandarin.

Table 1: Objective CER(%) for Mandarin systems

DATASET SIZE	18K	10K	7.5K	5K
BASELINE	4.65	9.40	18.33	FAIL
NLR	4.82	5.86	<b>7.14</b>	<b>13.64</b>
+WORD QUERY	<b>4.39</b>	<b>4.71</b>	7.28	14.78
w/o XLM-R	4.84	6.11	10.50	15.13

Table 2: Subjective error rate (%) on different test sets

DATASET SIZE	RARE		HETERONYMS	
	18K	10K	18K	10K
BASELINE	8.0	62.0	75.5	80.9
NLR	2.0	4.0	72.6	76.4
GENERAL				
DATASET SIZE	18K	10K	7.5K	5K
BASELINE	0.9	5.4	10.9	FAIL
NLR	0.3	1.9	4.1	8.5

such as Arabic and Indic ones share similar challenges.

To evaluate the method’s capability to leverage external knowledge when training data fail to cover necessary knowledge, we experimented on simulated low-resource cases with a single speaker Mandarin dataset of 18K samples downsampled to different sizes, without speaker or language embeddings. Besides, to evaluate the transferability of the lexicon-reading capability, we train a base model on a larger Mandarin dataset with 93k samples from 11 speakers, and adapt it to single-speaker Cantonese or Japanese data. The base model uses only speaker embeddings, so language embeddings are added to input embeddings and trained. All data are without word segmentation. Results are evaluated in comparison with baselines under the same setting but without Token2Knowledge layers. Other settings follow the prior work [5]. Waveforms are reconstructed from synthesized mel-spectrograms of a 100-sample held-out set on each language using the Griffin-Lim algorithm, quality from which sufficient for intelligibility evaluation. They are then sent to Azure Speech-to-Text to report objective character error rate (CER). Although the black-box recognizer has errors, it is generally reliable for comparing prevalence of pronunciation errors between systems [10]. A pretrained WaveNet is used to generate waveforms for subjective evaluations.

### 3.1. Mandarin experiments

Objective CERs are given in Table 1: when using the complete 18K-sample data, baseline and NLR models produce similar results, with CER both close to the ground truth one of 4.35%. But with fewer samples, NLR models are more resistant to reduced data and constantly outperform the baseline (which fails to converge at 5K data), showing that NLR successfully leverages pronunciation knowledge from lexicon texts when knowledge from data is insufficient. +Word Query and w/o XLM-R models will be discussed in Section 3.3.

Beside the general cases, we create test sets focused on scenarios with special challenges. First, for the capability to generalize to rare and even unseen characters (i.e. few-shot or zero-

Table 3: CER(%) for low-resource adaptation to a different language, with different dataset size for each column

CANTONESE	2K	1K	750	500	250
BASELINE	12.32	14.89	17.64	22.27	35.08
NLR	8.79	10.04	10.26	10.73	12.85
JAPANESE	5K	3K	2K	1K	750
BASELINE	12.46	15.99	18.66	26.85	33.76
NLR	10.50	12.48	13.95	19.29	21.81

shot cases), we create a 100-utterance RARE test set. Each of the utterances contains an uncommon character. They only appear once in the full 18K data, and 50 of them unseen in the 10K subset. To avoid interference, other used characters are simple to handle. 18K and 10K-data baseline and NLR models with the lowest objective CERs are used to synthesize the speeches to be inspected by a language expert. As shown in Table 2, both NLR models give correct pronunciations in most cases regardless of dataset sizes. Baselines show more errors to internalize the knowledge that is sparse in the training data, not to say the 10K case when such knowledge is unavailable. It also shows the extensibility of the framework that new knowledge (like pronunciations of unseen characters) can be directly introduced to a trained model. An experiment further supports this: we train a model with 10K data but remove lexicon entries for characters unseen in the 2K subset to simulate an incomplete lexicon, and the best CER can only reach 6.38%. But if the full lexicon is additionally given on evaluation, the CER reaches 5.91%, indicating that the model leverages and generalizes to new lexicon entries unseen during training.

Similarly, we create a rather challenging HETERONYM test set. We choose 115 characters with multiple readings. For each character, utterances are created using each reading except the most frequent one in training, hence a baseline that simply uses the most frequent reading will have 100% error rate. Total 292 utterances are used, and the error rates averaged by each reading of each character are reported. As shown in Table 2, NLR obtain improved ability regarding resolving heteronyms, showing an increased “soft” ability to decide the pronunciation according to the context, although the test remains challenging.

Furthermore, we evaluate subjective CERs on synthetic speech used to report Table 1. Given in Table 2 as the GENERAL CER, NLR constantly outperforms the baseline. We also find that low-resource cases errors in NLR are more due to misalignments, not mispronunciations. Besides, the curve is consistent with the objective metrics, showing the reliability to compare intelligibility with objective CER.

### 3.2. Cross-language transfer learning

Cantonese is a Chinese variant with systematic phonetic differences from Mandarin, making them mutually unintelligible. However, as shown by the results in Table 3, with only 2K Cantonese samples the CER of an adapted NLR is close to the ground truth (8.45%), and CERs increase slowly when using fewer samples. Japanese is more difficult: unlike Chinese, almost all characters have multiple readings. The native *kun-yomi* readings are not of Chinese origin, and are often multisyllabic which adds to the modeling difficulty. As a result, the CER gap between the 5K model and the ground truth (7.20%) is

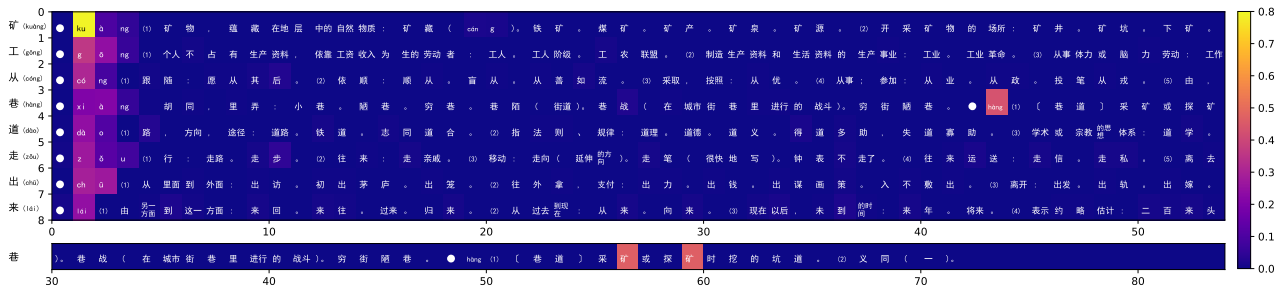


Figure 3: An alignment heatmap from Token2Knowledge attention with input tokens labelled on y-axis, and corresponding lexicon texts in the grid along x-axis, plus the row of the heatmap for “巷” in another head. Lexicon texts that are too long are omitted.

large. Nevertheless, the results are constantly improved with NLR, showing that the lexicon-reading capability is not fully language-dependent, and models trained on one language can be transferred to a distant language with fewer data.

### 3.3. Comparative study

We attempt to perform word segmentation on the texts and query the lexicon on a word-level granularity. In this way, the intonation and heteronym resolution might be easier. As shown by the +WORD QUERY model in Table 1, the performance is slightly improved with relatively rich data. However, it harms the performance with few data, possibly because representing readings of words but not monosyllabic characters is more difficult. Hence, word segmentation is generally not essential for NLR. We also tried to remove pretrained XLM-R, but use the same set of character embeddings to encode both inputs and lexicon text values. To capture higher-level semantic information, an additional transformer encoder layer is applied upon the embeddings to produce the keys. As shown in w/o XLM-R, limited regression can be observed in the results, particularly with fewer data, which shows that our method might be also applied to languages without such a pretrained model.

### 3.4. Case study

Figure 3 shows an alignment heatmap from the 18K-data Mandarin model synthesizing the given utterance. Pinyins of the correct pronunciations have a high alignment weight, indicating that the model extracts the correct pinyin from the lexicon entry, while irrelevant pinyins (like column 19 “cáng” in row 0 “矿”) are ignored. While the character “巷” is a heteronym, and under this context the character should be pronounced with an infrequent reading “hàng” which is only used in mining terms. The model produces the correct reading, which can be interpreted by the larger alignment weight assigned to “hàng” instead of “xiàng” in the lexicon entry. This is possibly because another head of attention for “巷” extracts context vectors from tokens “矿” (“mine”) in the lexicon entry, as shown in the lower half of the figure. More, the pronunciation can be directly manipulated by changing lexicon texts fed to the model for each character, such as by adding, modifying, or removing pronunciations in the text, as given in our audio samples, showing better controllability than ordinary end-to-end TTS or G2P models.

### 3.5. Naturalness evaluation

To evaluate the naturalness, we randomly sampled 50 utterances from the Mandarin held-out set. Mean opinion scores (MOS)

Table 4: Mean opinion score of different voices, with 95% confidence interval

DATASET SIZE	18K	10K
BASELINE	3.84±0.06	3.65±0.07
NLR	<b>3.85±0.06</b>	<b>3.80±0.06</b>
GROUND TRUTH	3.98±0.07	

are given by 20 judges per utterance. As shown in Table 4, our NLR model shows good naturalness, on par with the baseline with rich resources and better under lower resources.

## 4. Related work

Our work is closely related to Byte2Speech [5], as we follow their transformer-based model and scalability-centered goal, while their performances on non-phonemic scripts are unsatisfying. The framework is inspired by models that extract textual knowledge [7, 8], possibly from lexicon, for commonsense reasoning [11], QA [12], and Chinese sequence labelling [13]. Lexicon texts have been used in word sense disambiguation [14] and machine translation (MT) [15], while we apply the idea to TTS. The particular task we apply to is closely related to G2P and polyphone disambiguation with rich prior work, particularly on Chinese, Japanese, and Arabic [16, 17, 18, 19], possibly enhanced by explicit rules [20] or implicit external knowledge [21], and lexicons have been used during training [22, 23]. While the flexibility of using external knowledge has been discussed in MT [24] and G2P [25], though knowledge from raw lexicon texts were not leveraged. Representations from pretrained language models introduce additional knowledge and enhance TTS and G2P performances [26, 27, 28, 29, 30], but in most cases, the training data contain limited phonemic information, making these models less useful for capturing pronunciations. Also, knowledge from the neural models is lack of explicit representation and control.

## 5. Conclusion and future work

We demonstrate that knowledge can be extracted from external raw texts in end-to-end TTS models, with the application of Neural Lexicon Reader that obtains pronunciations from lexicon entries. Future work may include improving the extraction capability and applying it to other types of knowledge such as semantic meanings to improve speech prosody and naturalness.

## 6. References

- [1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Ryan, R. A. Saurous, Y. Agiomyriannakis, and Y. Wu, “Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions,” in *Proc. ICASSP 2018*. IEEE, 2018, pp. 4779–4783.
- [2] Y. Yasuda, X. Wang, and J. Yamagishi, “Investigation of learning abilities on linguistic features in sequence-to-sequence text-to-speech synthesis,” *Comput. Speech Lang.*, vol. 67, p. 101183, 2021.
- [3] J. Fong, J. Taylor, K. Richmond, and S. King, “A comparison between letters and phones as input to sequence-to-sequence models for speech synthesis,” in *The 10th ISCA Speech Synthesis Workshop*. ISCA, 2019, pp. 223–227.
- [4] J. Taylor and K. Richmond, “Analysis of pronunciation learning in end-to-end speech synthesis,” in *Proc. Interspeech 2019*. ISCA, 2019, pp. 2070–2074.
- [5] M. He, J. Yang, L. He, and F. K. Soong, “Multilingual Byte2Speech models for scalable low-resource speech synthesis,” *CoRR*, vol. abs/2103.03541, 2021.
- [6] F. Petroni, T. Rocktäschel, S. Riedel, P. S. H. Lewis, A. Bakhtin, Y. Wu, and A. H. Miller, “Language models as knowledge bases?” in *Proc. EMNLP/IJCNLP 2019*. ACL, 2019, pp. 2463–2473.
- [7] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, “Bidirectional attention flow for machine comprehension,” in *Proc. ICLR 2017*. OpenReview.net, 2017.
- [8] D. Chen, A. Fisch, J. Weston, and A. Bordes, “Reading Wikipedia to answer open-domain questions,” in *Proc. ACL 2017*. ACL, 2017, pp. 1870–1879.
- [9] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, “Unsupervised cross-lingual representation learning at scale,” in *Proc. ACL 2020*. ACL, 2020, pp. 8440–8451.
- [10] J. Taylor and K. Richmond, “Confidence intervals for ASR-based TTS evaluation,” in *Proc. Interspeech 2021*. ISCA, 2021, pp. 2791–2795.
- [11] L. Bauer, Y. Wang, and M. Bansal, “Commonsense for generative multi-hop question answering tasks,” in *Proc. EMNLP 2018*. ACL, 2018, pp. 4220–4230.
- [12] C. Wang and H. Jiang, “Explicit utilization of general knowledge in machine reading comprehension,” in *Proc. ACL 2019*. ACL, 2019, pp. 2263–2272.
- [13] W. Liu, X. Fu, Y. Zhang, and W. Xiao, “Lexicon enhanced Chinese sequence labeling using BERT adapter,” in *Proc. ACL/IJCNLP 2021*. ACL, 2021, pp. 5847–5858.
- [14] L. Huang, C. Sun, X. Qiu, and X. Huang, “GlossBERT: BERT for word sense disambiguation with gloss knowledge,” in *Proc. EMNLP-IJCNLP 2019*. ACL, 2019, pp. 3507–3512.
- [15] X. J. Zhong and D. Chiang, “Look it up: Bilingual and monolingual dictionaries improve neural machine translation,” *CoRR*, vol. abs/2010.05997, 2020.
- [16] J. Ni, Y. Shiga, and H. Kawai, “Multilingual grapheme-to-phoneme conversion with global character vectors,” in *Proc. Interspeech 2018*. ISCA, 2018, pp. 2823–2827.
- [17] M. Yu, H. D. Nguyen, A. Sokolov, J. Lepird, K. M. Sathyendra, S. Choudhary, A. Mouchtaris, and S. Kunzmann, “Multilingual grapheme-to-phoneme conversion with byte representation,” in *Proc. ICASSP 2020*. IEEE, 2020, pp. 8234–8238.
- [18] H. Zhang, H. Pan, and X. Li, “A mask-based model for Mandarin Chinese polyphone disambiguation,” in *Proc. Interspeech 2020*. ISCA, 2020, pp. 1728–1732.
- [19] S. Alqahtani, H. Aldarmaki, and M. T. Diab, “Homograph disambiguation through selective diacritic restoration,” in *Proc. Fourth Arabic Natural Language Processing Workshop, WANLP@ACL 2019*. ACL, 2019, pp. 49–59.
- [20] K. Gorman, G. Mazovetskiy, and V. Nikolaev, “Improving homograph disambiguation with supervised machine learning,” in *Proc. LREC 2018*. European Language Resources Association (ELRA), 2018.
- [21] D. Dai, Z. Wu, S. Kang, X. Wu, J. Jia, D. Su, D. Yu, and H. Meng, “Disambiguation of Chinese polyphones in an end-to-end framework with semantic features extracted by pre-trained BERT,” in *Proc. Interspeech 2019*. ISCA, 2019, pp. 2090–2094.
- [22] Y. Zou, L. Dong, and B. Xu, “Boosting character-based Chinese speech synthesis via multi-task learning and dictionary tutoring,” in *Proc. Interspeech 2019*. ISCA, 2019, pp. 2055–2059.
- [23] Y. Shi, C. Wang, Y. Chen, and B. Wang, “Polyphone disambiguation in Mandarin Chinese with semi-supervised learning,” in *Proc. Interspeech 2021*. ISCA, 2021, pp. 4109–4113.
- [24] Y. Feng, S. Zhang, A. Zhang, D. Wang, and A. Abel, “Memory-augmented neural machine translation,” in *Proc. EMNLP 2017*. ACL, 2017, pp. 1390–1399.
- [25] A. Bruguier, A. Bakhtin, and D. Sharma, “Dictionary augmented sequence-to-sequence neural network for grapheme to phoneme prediction,” in *Proc. Interspeech 2018*. ISCA, 2018, pp. 3733–3737.
- [26] T. Hayashi, S. Watanabe, T. Toda, K. Takeda, S. Toshniwal, and K. Livescu, “Pre-trained text embeddings for enhanced text-to-speech synthesis,” in *Proc. Interspeech 2019*. ISCA, 2019, pp. 4430–4434.
- [27] Y. Xiao, L. He, H. Ming, and F. K. Soong, “Improving prosody with linguistic and BERT derived features in multi-speaker based Mandarin Chinese neural TTS,” in *Proc. ICASSP 2020*. IEEE, 2020, pp. 6704–6708.
- [28] Y. Jia, H. Zen, J. Shen, Y. Zhang, and Y. Wu, “PnG BERT: Augmented BERT on phonemes and graphemes for neural TTS,” in *Proc. Interspeech 2021*. ISCA, 2021, pp. 151–155.
- [29] M. Nicolis and V. Klimkov, “Homograph disambiguation with contextual word embeddings for TTS systems,” in *The 11th ISCA Speech Synthesis Workshop*. ISCA, 2021, pp. 222–226.
- [30] R. Hida, M. Hamada, C. Kamada, E. Tsunoo, T. Sekiya, and T. Kumakura, “Polyphone disambiguation and accent prediction using pre-trained language models in Japanese TTS front-end,” in *Proc. ICASSP 2022*. IEEE, 2022, pp. 7132–7136.