Interspeech 2018
2-6 September 2018, Hyderabad

Multilingual grapheme-to-phoneme conversion with global character vectors

Jinfu Ni, Yoshinori Shiga, and Hisashi Kawai

Advanced Speech Technology Laboratory, ASTREC,
National Institute of Information and Communications Technology, Japan
{jinfu.ni, yoshinori.shiga, hisashi.kawai}@nict.go.jp

Abstract

Multilingual grapheme-to-phoneme (G2P) models are useful for multilingual speech synthesis because one model simultaneously copes with multilingual words. We propose a G2P model that combines global character vectors (GCVs) with bi-directional recurrent neural networks (BRNNs) and enables the direct conversion of text (as a sequence of characters) to pronunciation. GCVs are distributional, real-valued representations of characters and their contextual interactions that can be learned from a large-scale text corpus in an unsupervised manner. With the flexibility of learning GCVs from plain text resources, this method has an advantage: it enables monolingual G2P (MoG2P) and multilingual G2P (MuG2P) conversion.

We experiment in four languages (Japanese, Korean, Thai, and Chinese) with learning language-dependent (LD) and language-independent (LI) GCVs and then build MoG2P and MuG2P models with two-hidden-layer BRNNs. Our results show that both LD- and LI-GCV-based MoG2P models, whose performances are equivalent, achieved better than 97.7% syllable accuracy, which is a relative improvement from 27% to 90% depending on the language in comparison with Mecab-based models. As for MuG2P, the accuracy is around 98%, which is a slightly degraded performance compared to MoG2P. The proposed method also has the potential of the G2P conversion of non-normalized words, achieving 80% accuracy in Japanese.

Index Terms: G2P, BRNN, GloVe, text analysis, multilingual text-to-speech synthesis

1. Introduction

A grapheme is a character in Chinese or a number of characters in Thai that represent a sound (phoneme). The mapping between graphemes and phonemes is many-to-many. In text-to-speech (TTS) synthesis systems [1][2], grapheme-to-phoneme (G2P) models are critical because they describe the pronunciation of the input text. G2P models generate pronunciations, which are essentially based on the words, by referring to pronunciation dictionaries incorporated in the systems or by applying statistical/rule-based letter-to-phoneme conversion or their combination. Depending on particular languages, further efforts might be necessary to generate pronunciations, e.g., identifying phrase boundaries in Korean, determining the pronunciations of repetition signs in Thai, processing the devoicing of some vowels in Japanese, etc. If the input text is not normalized, such non-normalized words as numbers are read based on their context [3]. In the Japanese example shown in Fig. 1, for instance, the first "1" means "one", and the second "1" means "ten". Therefore, G2P conversion for TTS must consider a word's context.

To the best of our knowledge, in multilingual TTS systems, one G2P model basically works for one language to achieve high accuracy, although multilingual (Mu) G2P models exist

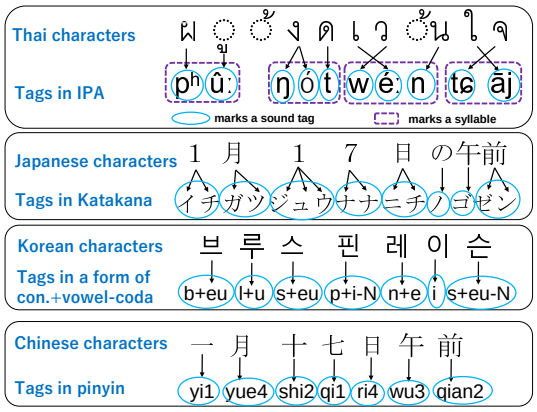

Figure 1: *Examples of grapheme and phoneme alignment*

for low-resource languages [4]. MuG2P models (one model for many languages) are useful for multilingual TTS since the input text may be mixed in two or more languages. A challenge to MuG2P is that some words are shared by several languages like Chinese and Japanese (Fig. 1). This problem may be alleviated by learning the pronunciations of words in context.

A statistical-based G2P is comprised of three steps:

- aligning training data: graphemes → phonemes [5];
- modeling: training a model by the training data [6]–[17];
- decoding: finding the most likely phonemes (pronunciation) given the model.

The task of aligning graphemes → phonemes is a problem of inducing links from graphemes to phonemes related by pronunciation [5]. The links from characters to phonemes (indicated by → in Fig. 1) may be nonlinear, as shown in Fig. 1 (cross links in Thai). Hereafter, the term "sound tag" (indicated by circles in Fig. 1) stands for a list of phonemes that is linked to a character. If no link exists, the character is denoted as a "null phoneme" by ϵ. As a result, alignment enables a character to have a sound tag that includes one or more phonemes or ϵ.

This paper describes a G2P model that combines global character vectors (GCVs) with bi-directional recurrent neural networks (BRNNs) [18]. GCVs are distributional, real-valued representations of characters and their contextual interactions, learned by GloVe [19] from a large-scale text corpus in an unsupervised manner. Multi-stack BRNNs are suitable for sequence modeling [20]. We show that GCV-based models outperform conventional Mecab-based models [21] and provide the flexibility to build multilingual G2P models and pronounce non-normalized words with rather good accuracy.

2. Related work

Using space vector models in TTS to represent word and letter types was previously proposed [23][24]. Both methods used

10.21437/Interspeech.2018-1626

matrix factorization methods with slim singular value decomposition (SVD) for generating a low-dimensional representation of letters [23]. We employed GloVe [19] to learn GCVs from a large-scale text corpus. GCVs directly capture the global corpus statistics based on a log-bilinear regression model.

A typical G2P approach uses joint n-gram models [6][7]. A weighted finite state transducer (WFST) is usually used to implement such a n-gram model [8]. In the literature, since G2P resembles a classification or machine learning problem, any of the following methods can be used: a maximum entropy classifier [9], a translation problem implemented in a sequence-to-sequence fashion [10], or a general machine learning problem with conditional random fields (CRF) [11][21] and HMM [13]. Neural network approaches are also used to solve G2P problems [14][15][10]. Hybrid models are generally effective. For example, Hahn et al. [16] combined n-gram models with decision tree models, and Wu et al. [17] combined them with CRF models. Rao et al. [15] combined a basic n-gram with BRNNs with a long short-term memory (LSTM) scheme. In the work, we combine GCVs with BRNNs to build G2P models that achieve high performance in multiple languages. GCVs implicitly represent n-gram models because they capture the global corpus statistics, including character interactions within a contextual window.

In contrast to our previous work [22], this work extends global syllable vectors (GSVs), which are language-dependent, to global character vectors (GCVs), which can be language-independent, and further explores monolingual G2P in multiple languages and multilingual G2P. In practice, using GCVs instead of GSVs no longer requires syllable breaking in Thai or word segmentation in Japanese.

3. Approach outline

Figure 2 diagrams GCV-BRNN-based G2P models, including model building (thin blue arrows) and G2P conversion (thick gray arrows). Model building consists of two steps. First, we learn a GCV table from a large-scale text corpus in an unsupervised manner using GloVe [19]. Second, we model the relationships between characters and sound tags by BRNNs in a supervised way, given an alignment of the training data between characters and sound tags. GCVs (encoding text by table search) are the input to BRNNs connected in the order of a stream of characters.

G2P conversion becomes the process of using GCVs to encode text by a table search followed by BRNN-based decoding GCVs to sound tags in a sequence-to-sequence manner.

3.1. GloVe-based learning GCVs

We used GloVe [19] to learn GCVs from a large-scale text corpus in an unsupervised manner. A large-scale text corpus gathers potential characters and their contextual interactions in the target languages. GloVe then uses the statistics of the character-character co-occurrences in the text corpus to learn the GCVs based on a global log-bilinear regression model [19], more particularly, by minimizing the following cost function [19]:

$$J = \sum_{i,j=1}^{N} q(x_{ij})(\boldsymbol{c}_i^T \tilde{\boldsymbol{c}}_j + b_i + \tilde{b}_j - log x_{ij})^2, \qquad (1)$$

where N is the number of unique characters, x_{ij} indicates the co-occurrence frequency of characters i and j, $q(x_{ij})$ is a weighting function to avoid the frequent characters overweighted [19], b_i and $\tilde{b}_j$ are biases, and $\boldsymbol{c}_i$ and $\tilde{\boldsymbol{c}}_j$ are space vectors to be learned. Note that $\boldsymbol{c}_i$ and $\tilde{\boldsymbol{c}}_i$ equivalently represent the ith character but with different initial values. As a result, $\boldsymbol{c}_i + \tilde{\boldsymbol{c}}_i$ is assigned to the ith character as its GCV.

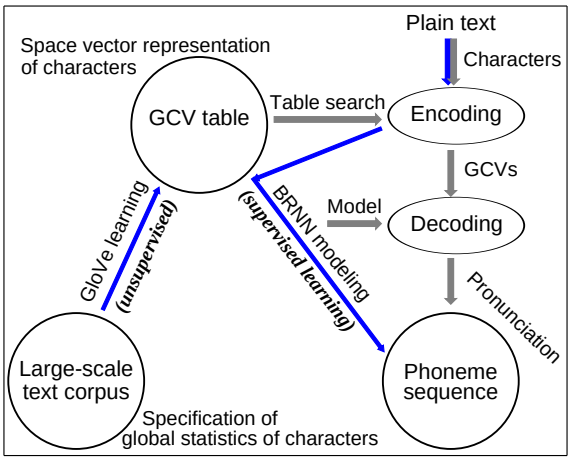

Figure 2: *Schematic diagram of G2P models combining GCVs learned from a large-scale text corpus with BRNNs*

We learn language-dependent (LD) GCVs and language-independent (LI) GCVs by setting up an appropriate text corpus. Hereafter, LDC stands for LD-GCV and LIC for LI-GCV.

3.2. BRNN-based decoding GCVs to sound tags

G2P conversion is performed by BRNN-based prediction from sequences of GCVs (encoding text) to sequences of sound tags. First, BRNNs are trained in a supervised way to learn the relationships between GCVs instead of characters and sound tags represented by one-hot vectors. We employed the standard multiclass cross-entropy as an objective function to train BRNNs with two hidden layers for this purpose, given an alignment of the training data between characters and sound tags. Eqs. (2) to (6) formally express the neural networks [20]:

$$\vec{h}_j^{(0)} = f(\vec{W}^{(0)} x_j + \vec{V}^{(0)} \vec{h}_{j-1}^{(0)} + \vec{b}^{(0)}) \qquad (2)$$

$$\overleftarrow{h}_j^{(0)} = f(\overleftarrow{W}^{(0)} x_j + \overleftarrow{V}^{(0)} \overleftarrow{h}_{j+1}^{(0)} + \overleftarrow{b}^{(0)}) \qquad (3)$$

$$\vec{h}_j^{(i)} = f(\vec{W}_{\rightarrow}^{(i)} \vec{h}_j^{(i-1)} + \vec{W}_{\leftarrow}^{(i)} \overleftarrow{h}_j^{(i-1)} + \vec{V}^{(i)} \vec{h}_{j-1}^{(i)} + \vec{b}^{(i)}) \qquad (4)$$

$$\overleftarrow{h}_j^{(i)} = f(\overleftarrow{W}_{\rightarrow}^{(i)} \vec{h}_j^{(i-1)} + \overleftarrow{W}_{\leftarrow}^{(i)} \overleftarrow{h}_j^{(i-1)} + \overleftarrow{V}^{(i)} \overleftarrow{h}_{j+1}^{(i)} + \overleftarrow{b}^{(i)}) \qquad (5)$$

$$\boldsymbol{y}_j = \text{softmax}(U_{\rightarrow} \vec{h}_j^{(2)} + U_{\leftarrow} \overleftarrow{h}_j^{(2)} + \boldsymbol{c}), \qquad (6)$$

where $1 \leq i \leq 2$ and arrows $\rightarrow$ and $\leftarrow$ respectively indicate forward and backward directions. $\boldsymbol{x}_j$ stands for the jth character's GCV, $j = 1, ..., n$ (the number of a sentence's characters), $\boldsymbol{h}$ is the hidden variables, $\boldsymbol{W}$, $\boldsymbol{V}$, and $\boldsymbol{U}$ are the weight matrices, $\boldsymbol{b}$ and $\boldsymbol{c}$ are the bias vectors, and $f(x)$ is the tanh function.

At the output layer, $\boldsymbol{y}_j$ gives the probabilities of all the sound tags for the jth character. A sound tag is then chosen for a character, basically using $argmax(\boldsymbol{y}_j)$, i.e., top-1.

4. Experimental setup

We evaluated our proposed method in Thai, Japanese, Korean, and Chinese, focusing on the four aspects below:

- effectiveness of LIC for G2P in multiple languages, including one LIC-BRNN model in each language and one LIC-BRNN model for two or three languages;

- equivalence of LIC- and LDC-based G2P models in performance;
- robustness of LIC-BRNN G2P when training data are limited;
- G2P conversion of non-normalized words.

Table 1 tabulates the datasets (including sentences and isolated words) used for the supervised training and the evaluation. The training, development, and test sets are disjointed. Additionally, we collected a test set (5k Japanese sentences) to evaluate G2P with non-normalized text. Every sentence has one or more non-normalized words (digit or alphabet strings).

Table 1: *List of training, development, and test sets*

Lang.	Training (sen.+word)	Devep. (s+w)	Test (s+w)
Thai	7.5k+38k	1k+1k	1k+1k
Japanese	65k+0	1.8k+0	1.5k+0
Korean	18k+93k	1k+1k	1k+1k
Chinese	44k+448k	3k+0	3.7k+0

4.1. GCV learning

We used a large-scale text corpus of about 500 million characters encoded in UTF-8 from the four languages to learn the LDC and LIC with a fixed 20-character window. We learned several GCVs with vector sizes of 50, 100, 200, 300, or 512. There were 13,700 distinctive characters in total.

4.2. Alignment between characters and sound tags

The datasets (corpus) in Table 1 had word-level pronunciation checked by native speakers. An alignment of the words between characters and sound tags is needed for training the BRNNs. In Chinese and Korean, this step is straightforward since a character has a sound. In Thai and Japanese, constraint-based alignment [14][25] is used in the following semi-automatic and interactive manner:

- Construct sets of sound tags for characters and count the occurrences of character-sound tags in the corpus.
- Align a word between characters and sound tags by building a tree with nodes as character-sound tags.
  - Grow a tree using the sets of character-sound tags.
  - Prune it using the word's sound tags.
  - Define a new character-sound tag and add it to the sets of sound tags if a tree can't be built.
- Select such a path as the alignment that maximizes the sum of occurrences of the word's sound tags.

We obtained 623 distinctive sound tags in Thai, 2197 in Japanese, 1916 in Korean, and 1452 in Chinese. To restore the phonemes after the G2P conversion and determine the syllables from a sequence of phonemes in Thai, we added labels to the consonants to indicate their positions (first, next, and last) in a syllable and the vowels with cross links (Fig. 1). All the tone marks take sound tag ϵ in our experiments.

The sets of sound tags are not optimal because the constraint-based alignment is simplified. Further improvement is possible.

Table 2: *Results of GCV-BRNN-based G2P conversion evaluated in character-to-sound-tag accuracy (%) with GCV size 300 for MoG2P and 512 for MuG2P: OOT indicates out of sound tags that were not used in the language (in percentage terms).*

Lang.	MoG2P		MuG2P-b		MuG2P-t	
	LDC	LIC	LIC	OOT	LIC	OOT
Japanese	**98.77**	98.74	98.05	0.001	n/a	
Chinese	99.32	**99.37**	98.69	0.008	98.83	0.025
Korean	97.74	**97.94**	n/a		97.50	0.018
Thai	**99.18**	99.16			98.97	0.0

Table 3: *Comparison of GCV-based G2P with Mecab-based G2P by relative improvements in syllable and word error rates: MoG2P models were used in evaluation.*

Lang.	Syllable error rate (%)			Word error rate (%)		
	Mec-1	LIC-1	Improve	Mec-1	LIC-1	Improve
Thai	3.89	**2.00**	*48.50%*	3.97	**1.73**	*56.51%*
Japanese	2.09	**1.31**	*37.28%*	2.05	**1.68**	*17.78%*
Korean	24.2	**2.25**	*90.69%*	29.5	**2.88**	*90.25%*
Chinese	0.82	**0.63**	*27.0%*	1.26	**0.91**	*27.27%*

4.3. Network training

We trained the BRNNs with the following hyperparameters:

- number of units of input layers: GCV size;
- number of units of output layers: number of sound tags;
- number of hidden layers: 2;
- number of hidden units: 50, 100, 150, or 200;
- used a stochastic gradient descent with a fixed momentum (0.9) with a small learning rate (0.0001);
- size of a mini-batch: 20 samples (sentences or words);
- maximum epoch: 2000.

We used the models with the best performance for the development set as the final models to be evaluated.

4.4. Mecab-based baseline for comparison

Mecab [21], a CRF-based morphological analyzer, is widely used in TTS for the morphological analysis of text and G2P by referring to a dictionary [1]. For comparison, we chose Meacb as baseline, partly because our training datasets have part-of-speech tags and word pronunciation suited for training Mecab-based G2P models for high performance. We used the same datasets in Table 1 to build Mecab-based G2P models.

4.5. List of G2P model symbols

We trained the following G2P models in our experiments:

- MoG2P: monolingual G2P each with LDC and LIC;
- MuG2P-b: bi-lingual G2P in Japanese and Chinese;
- MuG2P-t: tri-lingual G2P in Chinese, Korean, and Thai;
- Mec-x: Mecab-based G2P models trained by the samples with $x\times$ the size of the datasets;
- LIC-x: LIC-MoG2P models trained by the same samples as used in training Mec-x.

x takes 1 for all the four languages but $\frac{1}{2}$, $\frac{1}{4}$, and $\frac{1}{8}$ in Thai, Japanese, and Korean. Note that the same development and test sets were used throughout the experiments.

Table 4: *Results of MoG2P conversion of non-normalized text: Japanese LIC-MoG2P in Table 2 was used in test.*

Japanese	Number of words/word error rate (%)
Non-normalized	13662/*20.53*
Common words	69458/1.958
Total words	83120/5.012

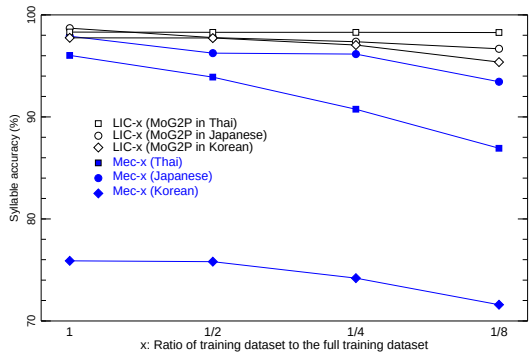

Figure 3: *Illustration of robustness of GCV-based models with limited training data in comparison with Mecab-based models*

5. Results

The experimental results are shown in Tables 2–4 and Figs. 3–5. BRNNs have 100 hidden units, and the GCV size is 300, except when clearly noted otherwise in the item description.

The following observations are based on our results:

- MoG2P models achieved very high performance: 99.18% character-to-sound tag accuracy in Thai, 98.77% in Japanese, 97.74% in Korean, and 99.32% in Chinese (Table 2). Compared to the Mecab-based method, our proposed method got significant relative improvement in the syllable error rates from 27% to 90%, depending on the individual language, and from 17.7% to 90% in the word error rates (Table 3).
- The MuG2P models also achieved high performance, more than 97.5% character-to-sound tag accuracy in all four languages, but it was slightly degraded compared to MoG2P (Table 2). The degradation was caused by the large size of the BRNN outputs: 3721 units in MuG2P-t and 3401 in MuG2P-b. Even though we changed the number of hidden units from 100 to 150, unfortunately, no significant improvement was obtained.
- A few out of sound tags (OOTs) exist in MuG2P (Table 2). However, for MuG2P-t (Chinese, Korean, and Thai), the OOTs are less than 0.03% depending on the language. In the case of MuG2P-b (Japanese and Chinese), the OOTs are less than 0.008%. These are very positive results. In fact, some words and even isolated sentences are not distinctive in Chinese and Japanese. OOT issues can be handled by choosing sound tags from the top-n based on individual languages, instead of the top-1, as in the experiments.
- The GCV-based G2P is more robust than Mecab-based G2P, especially when the training data are limited (Fig. 3). The latter refers to pronunciation dictionaries, and thus it is sensitive to the words available in the training.
- LIC- and LDC-based G2P models have equivalent performances (Fig. 4). In model training, some hyperparameters can be fixed. 100 hidden units are adequate for both LDC- and LIC-BRNN-based G2Ps (Fig. 5). As for the GCV size, our experiments indicate that 300 is adequate for MoG2P models, but 512 is better for relatively easy training of the MuG2P models.

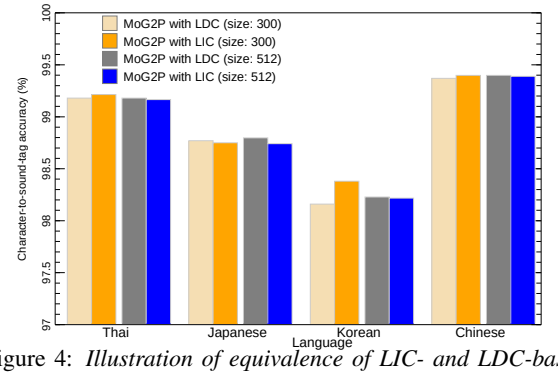

Figure 4: *Illustration of equivalence of LIC- and LDC-based G2P in performance using BRNNs with 100 hidden units*

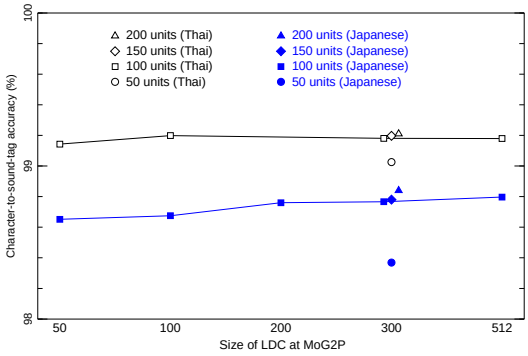

Figure 5: *Effects of varied GCV size and number of hidden units on resultant MoG2P performance*

- Around 80% word accuracy is achieved by the MoG2P conversion of non-normalized words in Japanese (Table 4). An informal analysis of the results indicates that G2P for numbers representing dates (year, month, and day) are basically correct. Frequent mistakes are related to unseen abbreviations (e.g., SPAM) and some numbers longer than three characters where the G2P conversion sometimes omits "thousand" or "hundred" in the middle of the numbers.

We believe that the performance can be further improved using more training samples in mixed languages and sufficient non-normalized words in context. Further work is needed.

6. Conclusions

We showed that, as features, GloVe-based global character vectors (GCVs) successfully achieved G2P conversion with deep learning in multiple languages. GCVs, either language-dependent or language-independent, can be learned from plain text corpora in an unsupervised way. The proposed method outperformed the conventional Mecab-based method and has the flexibility to implement a multilingual G2P to enable one model to cope with several languages. This is motivated by a desire to relatively easily implement multilingual text-to-speech [26].

Future work will improve our multilingual G2P models with non-normalized samples and evaluate our proposed method on common data for comparison with other approaches.

7. References

[1] H. Kawai, *et al.*, "XIMERA: A concatenative speech synthesis system with large scale corpora," *IEICE Trans. Inf. & Syst.,*, vol. J89-D, no. 12, pp. 2688-2698, 2006.

[2] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, pp. 7962-7966, 2013.

[3] R. Sproat and N. Jaitly, "An rnn model of text normalization," in *INTERSPEECH*. 2017.

[4] A. Deri and K. Knight, "Grapheme-to-phoneme models for (almost) any language," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 299-408, 2016.

[5] S. Jiampojamarn and G. Kondrak, "Letter-phoneme alignment: an exploration," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 780-788, 2010.

[6] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communications*, vol. 50, no. 5, pp. 434-451, 2008.

[7] L. Galescu and J. F. Allen, "Pronunciation of proper names with a joint n-gram model for bi-directional grapheme-to-phoneme conversion," in *INTERSPEECH*, 2002.

[8] J. R. Novak, P. R. Dixon, N. Minimatsu, K. Hirose, C. Hori, and H. Kasioka, "Improving wfst-based g2p conversion with alignment constraints and rnnlm n-best rescoring," in *INTERSPEECH*, 2012.

[9] S. F. Chen, "Conditional and joint models for grapheme-to-phoneme conversion," in *INTERSPEECH*, 2003.

[10] K. Rao and G. Zweig, "Sequence-to-sequence neural net models for grapheme-to-phoneme conversion," in *INTERSPEECH*, 2015.

[11] D. Wang and S. King, "Letter-to-sound pronunciation prediction using conditional random fields," *IEEE Signal Processing Letters*, vol. 18 (2), pp. 122-125, 2011.

[12] P. Lehnen, A. Allauzen, T. Lavergne, F. Yvon, S. Hahn, and H. Ney, "Structure learning in hidden conditional random fields for grapheme-to-phoneme conversion," in *INTERSPEECH*, 2013.

[13] S. Jiampojamarn, C. Cherry, and G. Kondark, "Joint processing and discriminative training for letter-to-phoneme conversion," in *Proceedings of ACL*, pp. 905-913, 2008.

[14] K. J. Jensen and S. Riis, "Self-organizing letter code-book for text-to-phoneme neural network model," in *INTERSPEECH*, pp. 318-321, 2000.

[15] K. Rao, F. Peng, H. Sak, and F. Beaufays, "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks," in *ICASSP*, 2015.

[16] S. Hahn, P. Vozila, and M. Bisani, "Comparison of grapheme-to-phoneme methods on large pronunciation dictionaries and lvcsr tasks," in *INTERSPEECH*, 2012.

[17] K. Wu, C. Allauzen, K. Hall, M. Riley, and B. Roark, "Encoding linear models as weighted finite-state transducers," in *INTERSPEECH*, 2014.

[18] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, 45(11), pp. 2673-2681, 1997.

[19] J. Pennington, R. Socher, and C. D. Manning, 2014, "GloVe: Global vectors for word representation," *http://nlp.stanford.edu/projects/glove/*.

[20] O. Irsory and C. Cardie, "Opinion mining with deep recurrent neural networks," in *Proc. the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 720-728, 2014.

[21] MeCab : Yet another part-of-speech and morphological analyzer. http://taku910.github.io/mecab/

[22] J. Ni, Y. Shiga, and H. Kawai, "Global syllable vectors for building tts front-end with deep learning," in *INTERSPEECH*, 2017.

[23] O. Watts, "Unsupervised learning for text-to-speech synthesis," *Ph.D. dissertation*, University of Edinburgh, 2012.

[24] H. Lu, S. King, and O. Watts, "Combining a vector space representation of linguistic context with a deep neural network for text-to-speech," in *Proc. the 8th ISCA Speech Synthesis Workshop (SSW)*, pp. 281-285, 2013.

[25] A. W. Black, K. Lenzo, and V. Pagel, "Issues in building general letter to sound rules," in *The Third ESCA Workshop in Speech Synthesis*, pp. 77-80, 1998.

[26] Y. Shiga and H. Kawai, "Multilingual speech synthesis system," *Journal of the National Institute of Information and Communications Technology*, vol. 59, nos 3/4, 2012.