Interspeech 2018
2-6 September 2018, Hyderabad

Acoustic-dependent phonemic transcription for text-to-speech synthesis

Kévin Vythelingum[1,2], *Yannick Estève*[2], *Olivier Rosec*[1]

[1]Voxygen, Pleumeur-Bodou, France
[2]LIUM, Le Mans University, France

{kevin.vythelingum,yannick.esteve}@univ-lemans.fr,
{kevin.vythelingum,olivier.rosec}@voxygen.fr

Abstract

Text-to-speech synthesis (TTS) purpose is to produce a speech signal from an input text. This implies the annotation of speech recordings with word and phonemic transcriptions. The overall quality of TTS highly depends on the accuracy of phonemic transcriptions. However, they are generally automatically produced by grapheme-to-phoneme conversion systems, which do not deal with speaker variability. In this work, we explore ways to obtain signal-dependent phonemic transcriptions. We investigate forced-alignment with enriched pronunciation lexicon and multimodal phonemic transcription. We then apply our results on error detection of grapheme-to-phoneme conversion hypotheses in order to find where the phonemic transcriptions may be erroneous. On a French TTS dataset, we show that we can detect up to 90.5% of errors of a state-of-the-art grapheme-to-phoneme conversion system by annotating less than 15.8% of phonemes as erroneous. This can help a human annotator to correct most of grapheme-to-phoneme conversion errors without checking a lot of data. In other words, our method can significantly reduce the cost of high quality TTS data creation.

Index Terms: grapheme-to-phoneme conversion, text-to-speech synthesis, automatic error detection, multimodal phonemic transcription, forced-alignment

1. Introduction

Text-to-speech synthesis (TTS) purpose is to produce a speech signal from an input text. To build such a system it is necessary to create a speech database (SDB) with text transcription. A voice talent is recorded on the reading of selected texts, which are also transcribed into phonemes. Besides, the speech is segmented in smaller acoustic units described by phonemes. Then, two paradigms could be distinguished for TTS: unit selection speech synthesis, where the synthesized speech signal comes from the selection and the concatenation of acoustic units [1], and statistical parametric speech synthesis, where acoustic features of the speech signal are predicted from a sequence of words [2]. SDBs are either used as acoustic units selection corpora, or as training corpora for acoustic models. In both cases, phonemic transcription should be very accurate to give satisfying results. Recently, works showed that many components of TTS systems can be replaced by neural networks models, using phonemic descriptors [3, 4, 5] or not [6, 7]. Accurate phonemic transcriptions are thus still mandatory for TTS components.

Phonemic transcriptions of SDBs are generally obtained by automatic grapheme-to-phoneme (G2P) conversion of text transcriptions. This can be done at word-level, where words are considered as isolated, or at sentence-level where lexical context is taken into account. Popular approaches for word-level G2P conversion are dictionary look-up and joint-sequence models [8, 9]. Rule-based [10], statistical machine translation [11] and sequence-to-sequence neural networks models [12, 13] apply both for word-level and sentence-level G2P conversion.

However, G2P conversion depends only on text transcription and cannot be adapted automatically to what speakers pronounce. This is an issue in SDBs annotation, where phonemic transcriptions should be manually reviewed to match correctly the speech signal. French TTS quality is indeed increased in [14] when phonemic transcriptions are corrected manually. Moreover, in [15], authors observed improvements of speech synthesis when phonemic transcriptions are more accurate. Actually, it is essential to obtain accurate phonemic transcriptions, which match the speech signal.

In a previous work, we showed we can detect the errors made by a commercial rule-based G2P conversion system by comparing the phonemic transcriptions hypotheses produced by a neural G2P conversion system to a forced alignment between the speech signal and a pronunciation lexicon [16]. Besides, we observed the neural G2P conversion hypotheses were more accurate than the ones produced by the rule-based G2P conversion system. In this work, we explore different ways to obtain signal-dependent phonemic transcriptions. We improve the acoustic models used in forced alignment, and investigate the use of end-to-end monomodal and multimodal phoneme recognition hypotheses. We then evaluate our models to the error detection task, applied to the neural G2P conversion hypotheses, our best model for phonemic transcription.

The paper is organized as follows. In section 2, we present the different phonemic transcription systems. Then, we describe the error detection task in section 3. Finally, we give the results obtained in both phonemic transcription and error detection tasks.

2. Phonemic transcription systems

We investigate how to obtain phonemic transcriptions from different modality (speech signal, text) available in TTS datasets. Our goal is to get the most accurate transcriptions according to what speakers pronounced during speech recording.

2.1. Joint-sequence model for grapheme-to-phoneme conversion

A joint-sequence model is a statistical model for sequences of *(graphemes, phonemes)* pairs [8, 9]. During the training stage, sequences of graphemes are aligned to sequences of phonemes. Then, a language model is trained on the resulting sequences of *(graphemes, phonemes)* pairs. At the decoding stage, possible sequences of phonemes are first generated from the input sequences of graphemes, and then the language model determines the most likely sequence of *(graphemes, phonemes)* pairs, and so the best sequence of phonemes.

We were using the Phonetisaurus toolkit [17, 18] and a

10.21437/Interspeech.2018-1306

6-gram SRILM language model [19, 20] to build the joint-sequence model. This model deals with isolated words only, as it does not model lexical context. That is why we do not generate directly with it phonemic transcription hypotheses for SDB annotation. However, we can enrich a pronunciation lexicon with its hypotheses and use it for forced alignment.

2.2. Neural sequence-to-sequence model for grapheme-to-phoneme conversion

G2P conversion systems infer phonemic sequences from text. However, it is sometimes impossible to choose the right pronunciation of a word only with its spelling. That is why we built a context-dependent G2P conversion system to take decisions with additional knowledge.

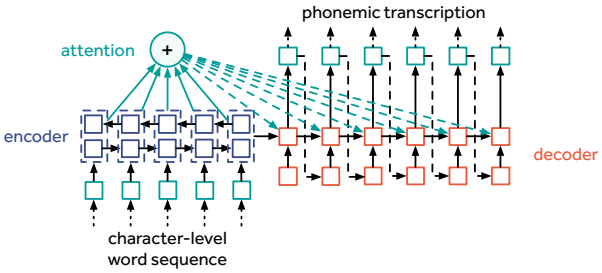

Figure 1: *Architecture of neural sequence-to-sequence grapheme-to-phoneme conversion model.*

We developed a G2P conversion system based on sequence-to-sequence neural network modeling. The system takes as inputs sentence-level sequences of characters and outputs sequences of phonemes. We chose to follow the encoder-decoder architecture described in [21] as we dealt with long sequences with diffent lengths in both input and output (see figure 1). We used the open-source neural machine translation toolkit *NMTPy* [22] to implement our models. The decoder is composed by two gated recurrent unit (GRU) layers interleaved with attention mechanism and the hidden state of the decoder is initialized with a non-linear transformation applied to the mean bidirectional encoder state. We use 64-dimensional embeddings and 128-dimensional hidden layers. During training, we use dropout with probability 0.4 after each recurrent layer. We also use the Adam optimization algorithm with a batch size of 32 and a learning rate of 10^{-4}.

2.3. Acoustic forced alignment for phonemic transcription

Another way to disambiguate word pronunciations is to exploit the speech signal as it is available in TTS datasets. For this purpose, we align the text transcription and the speech signal of SDBs at the phoneme level using an acoustic model and a pronunciation lexicon containing all the words in the transcriptions.

The acoustic models are trained with the Kaldi speech recognition toolkit [23]. First, we trained a GMM/HMM acoustic model on mel-frequency cepstrum coefficients (MFCC) features with feature space maximum likelihood linear regression (fMLLR) speaker adaptation. Then, we trained a TDNN-LSTM/HMM acoustic model [24] with 300-dimensional hidden layers based on the senone alignment from the GMM/HMM model. The model is taking as inputs 40-dimensional high-resolution MFCC features and 100-dimensional i-vectors.

The lexicon is first built by applying a commercial rule-based G2P system on the list of all words of the dataset. As the G2P conversion is processed without lexical context, several pronunciation hypotheses for each word are given. However, some pronunciations alternatives might still miss for some words. That is why we enriched the lexicon by adding n-best hypotheses from the joint-sequence model described in section 2.1. We tried different number of additional pronunciation hypotheses to build the pronunciation lexicons, and we processed the forced alignment with the different enriched lexicons. By increasing gradually the number of additional pronunciation alternatives, we attempt to show the impact of missing pronunciations in lexicons for forced alignment phonemic transcription.

2.4. End-to-end phoneme recognition

Although forced alignment is based on an acoustic model to choose which pronunciation of a lexicon is matching the best the speech signal, this approach still depend strongly on text and G2P conversion performances. In order to obtain a phonemic transcription hypothesis which depend only on the acoustic signal, we chose to investigate an end-to-end phoneme recognition model. We experimented the Deepspeech 2 architecture [25] which consists in a neural network with two convolutional layers and five 800-dimensional bidirectionnal GRU layers. The model was trained with CTC (Connectionist Temporal Classification) function [26] to avoid the need to align the speech signal and the sequences of phonemes. This mitigates the effect of alignment errors in training.

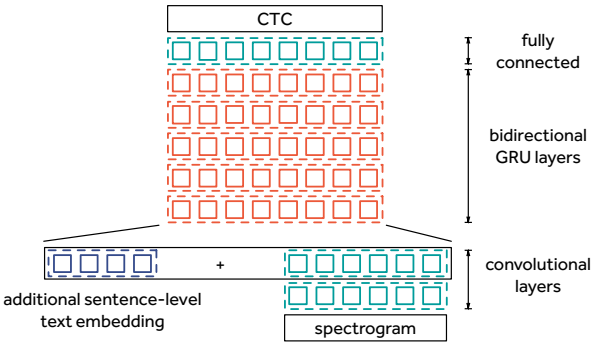

Figure 2: *Architecture of end-to-end phoneme recognition model with additional sentence-level text embedding.*

We also investigated the addition of features computed from text transcriptions. Indeed, it can help the phoneme recognition to have a knowledge about the text transcription during decoding, in addition to the speech signal. We computed sentence-level text embeddings with the neural G2P conversion model described in section 2.2 in two manners:

1. for each sentence, we took the last hidden state of the encoder as a 256-dimensional vector
2. for each sentence, we took the sum of the hidden states of the encoder as a 256-dimensional vector

As a bidirectional encoder, (1) is supposed to contain the history of the whole sentence, and (2) is like a attention mechanism with the same weight for each hidden state. After extracting the embeddings for each sentence, we concatenated these vectors to the input of the first GRU layer of the end-to-end phoneme recognition model (see figure 2).

3. Application to error detection of grapheme-to-phoneme conversion

We obtained contrastive phonemic transcriptions hypotheses from different sources. They can be infered from speech only, from text only or from a combination of the two. We apply now these results to detect the errors made by automatic grapheme-to-phoneme conversion, especially the ones made by our neural sequence-to-sequence G2P conversion model. This will allow a human annotator to improve quickly phonemic transcription quality by correcting only the TTS dataset parts detected as erroneous.

First, we compare the outputs of the neural sequence-to-sequence G2P conversion model to a manually transcribed reference. This allow us to label each phoneme of the hypothesis as correct or erroneous: this annotation is considered the reference for error detection. Then, we compare phonemic transcription hypotheses from forced alignment and from end-to-end phoneme recognition to the hypotheses from the neural G2P conversion model: we put *error* labels where phonemes are mismatching and *correct* labels where phonemes are the same. This gives the error detection hypothesis. Our goal is to retrieve with the error detection hypothesis where the *error* labels are in the error detection reference.

4. Results

4.1. Evaluation data

We trained our models using internal French TTS datasets containing approximately 50 hours of speech data from 9 speakers segmented into 90,135 utterances. The results are then given by testing our models on internal French TTS datasets containing approximately 1 hour of speech data from 3 speakers segmented into 951 utterances. All the speech data is transcribed manually at both word and phoneme level. Even if we can expect better results by training models on data similar to our use-case, it is possible to use any corpus prepared for ASR development to train the acoustic models.

4.2. Phonemic transcription evaluation

We first evaluate our models on the phonemic transcription task, with the Phone Error Rate (PER) metric. This metric, generally used in G2P conversion and phoneme recognition, gives the mean percentage deviation in Levenshtein distance with the manually corrected phonemic transcription.

We evaluate phonemic transcription at sentence-level. Table 1 gives results for G2P conversion systems, table 2 gives results for forced alignment with different pronunciation lexicon, and table 3 gives results for end-to-end speech recognition.

#	System	PER
(S0)	joint-sequence model	9.2
(S1)	neural sequence-to-sequence model	**2.8**

Table 1: *Phone error rate (%) for sentence-level G2P conversion*

Neural sequence-to-sequence model outperforms joint-sequence model for sentence-level G2P conversion. Indeed, joint-sequence model does not take into account lexical context and thus the pronunciation hypotheses of words is the same for every occurrence.

#	System	PER
(S2)	acoustic forced alignment	4.6
(S2+1)	(S2) + 1-best (S0) hypotheses	**4.3**
(S2+2)	(S2) + 2-best (S0) hypotheses	4.9
(S2+3)	(S2) + 3-best (S0) hypotheses	5.3

Table 2: *Phone error rate (%) for sentence-level acoustic forced alignment*

We see in table 2 that adding the 1-best G2P hypotheses of the joint-sequence model in the forced alignment lexicon benefits to phonemic transcription. However, the phone error rate increases when we add more than one pronunciation alternative. Moreover, the best result is obtained by adding the output given by the neural sequence-to-sequence model.

#	System	PER
(S3)	end-to-end phoneme recognition	9.9
(S3+last)	(S3) + last hidden state of (S1)	10.3
(S3+sum)	(S3) + hidden states sum of (S1)	**9.6**

Table 3: *Phone error rate (%) for sentence-level end-to-end phoneme recognition*

End-to-end phoneme recognition accuracy is lower than G2P conversion and forced alignment phonemic transcription. However, we observe that it improves results to concatenate to the input of the recurrent layers the hidden states sum of the neural sequence-to-sequence model encoder. An information on the text sequence can thus be managed by the network to obtain more accurate phonemic transcription.

4.3. Error detection evaluation

We then evaluate error detection results with *Precision*, *Recall* and *Manual Checking Rate (MCR)* metrics. Precision and Recall are standard metrics to evaluate error detection systems. They indicate respectively the proportion of true alarms raised by the error detection system and the proportion of detected errors. We introduce also the Manual Checking Rate (MCR), which shows the amount of data which is annotated as erroneous by the system:

$$MCR = \frac{number\ of\ phonemes\ annotated\ as\ erroneous}{number\ of\ phonemes\ in\ the\ reference}$$

In other words, the MCR indicates how much phonemes should be checked by a human annotator to correct the amount of errors given by Recall. We want to maximize Precision and Recall while we want to minimize MCR.

First, we experiment error detection with forced alignment using different pronunciation lexicons. We use the baseline lexicon, and then add progressively to this lexicon the one, two and three bests hypotheses of the joint-sequence model. This allows us to cover more pronunciation variants in the lexicon and thus gives more liberty to the acoustic model. Table 4 gives the results in terms of Precision, Recall and MCR.

We observe the best Precision and MCR are obtained with the addition of the one best G2P hypotheses of the joint-sequence model in the forced alignment lexicon. This mean with this configuration we can detect 68.9% of G2P errors by checking manually only 5.1% of the dataset. However, if we want to correct more errors, we can enrich the forced alignment

#	System	Precision	Recall	MCR
(0)	(S2)	35.0	67.8	5.3
(1)	(S2+1)	**36.7**	68.9	**5.1**
(2)	(S2+2)	34.1	75.2	6.0
(3)	(S2+3)	32.4	**77.3**	6.5

Table 4: *Evaluation of error detection with forced alignment*

lexicon more and then detect up to 77.3% of errors. This requires checking manually 6.5% of the dataset.

Table 5 gives the evaluation results of G2P conversion error detection with end-to-end phoneme recognition. The model (4), which only takes into account the speech signal, obtained the best Recall of the three models with 81.8%. However, if we need to check less than 12% of the dataset, the model (6) can detect 79.6% of G2P conversion errors, which is more than what we have with forced alignment.

#	System	Precision	Recall	MCR
(4)	(S3)	18.2	**81.8**	12.2
(5)	(S3+last)	18.1	80.7	12.1
(6)	(S3+sum)	**18.0**	79.6	**12.0**

Table 5: *Evaluation of error detection with end-to-end phoneme recognition*

Finally, we combined forced alignment and end-to-end phoneme recognition for error detection by comparing the contrastive hypotheses of the two systems to the sentence-level G2P conversion hypotheses. Table 6 shows the error detection results obtained with this combination.

#	System	Precision	Recall	MCR
(7)	(0)+(4)	15.5	87.2	15.2
(8)	(1)+(4)	15.7	87.4	15.0
(9)	(2)+(4)	15.7	89.7	15.5
(10)	(3)+(4)	15.5	90.2	15.8
(11)	(0)+(5)	15.6	87.4	15.2
(12)	(1)+(5)	**15.8**	87.2	**14.9**
(13)	(2)+(5)	15.7	89.7	15.5
(14)	(3)+(5)	15.5	**90.5**	15.8
(15)	(0)+(6)	15.4	86.3	15.2
(16)	(1)+(6)	15.6	86.5	15.0
(17)	(2)+(6)	15.5	88.7	15.5
(18)	(3)+(6)	15.3	89.5	15.9

Table 6: *Evaluation of error detection with combination of forced alignment and end-to-end phoneme recognition*

The error detection task benefits more from the combination between forced alignment and end-to-end phoneme recognition with the last hidden state of neural G2P conversion system encoder as external embedding. Indeed, we obtain the best Precision and MCR with combination (12), where we get 15.8% of Precision and 14.9% of MCR, and we obtain the best Recall with combination (14), where we have with 90.5% the maximum error coverage of all systems.

In short, depending on the trade-off we want between Recall and MCR, we need to choose between only forced alignment, which gives the best MCR, only phoneme recognition, which gives average results, or a combination of forced alignment and phoneme recognition, which gives the best Recall. Besides, sentence-level text embeddings of both types used during phoneme recognition gives interesting results for error detection.

5. Conclusion

We investigated different ways to obtain phonemic transcription of TTS datasets when speech signal and text are available. We compared joint-sequence modeling and neural sequence-to-sequence grapheme-to-phoneme conversion, forced alignment with enriched pronunciation lexicon, and end-to-end phoneme recognition. As we wanted to benefit from both speech signal and text as the same time for phonemic transcription, we showed we can successfully extract sentence-level embeddings from sequence-to-sequence grapheme-to-phoneme conversion model and use them during phoneme recognition. Finally, we apply our different models to the task of detecting grapheme-to-phoneme conversion errors. With the combination of the contrastive phonemisation hypotheses we obtained, we showed on a French TTS dataset we can detect up to 90.5% of errors by labelling only 15.8% of phonemes as doubtful. This means a human annotator can improve drastically the quality of a TTS corpus in a limited amount of time, by correcting only the dataset parts annotated as erroneous. Further work will consist in exploring other use cases for multimodal phonemic transcription and grapheme-to-phoneme error detection.

6. References

[1] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1996, pp. 373–376.

[2] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," in *Speech Communication*, 2009, pp. 1039–1064.

[3] A. Van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *arXiv:1609.03499v2*, 2016.

[4] S. Ark, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, S. Sengupta, and M. Shoeybi, "Deep voice: Real-time neural text-to-speech," in *arXiv:1702.07825v2*, 2017.

[5] S. Ark, G. Diamos, A. Gibiansky, J. Miller, K. Peng, W. Ping, J. Raiman, and Z. Y., "Deep voice 2: Multi-speaker neural text-to-speech," in *arXiv:1705.08947v1*, 2017.

[6] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *arXiv:1703.10135*, 2017.

[7] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *arXiv:1712.05884*, 2017.

[8] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," in *Speech Communication*, 2008, pp. 434–451.

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

[10] F. Béchet, "Lia phon : un système complet de phonétisation de textes," in *Traitement Automatique des Langues (TAL)*, 2001, pp. 47–67.

[11] A. Laurent, P. Delglise, and S. Meignier, "Grapheme to phoneme conversion using an smt system," in *Proceedings of InterSpeech*, 2009.

[12] K. Rao, F. Peng, H. Sak, and F. Beaufays, "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4225–4229.

[13] K. Yao and G. Zweig, "Sequence-to-sequence neural net models for grapheme-to-phoneme conversion," in *Proceedings of InterSpeech*, 2015.

[14] S. Brognaux, P. B., T. Drugman, and L. D., "Speech synthesis in various communicative situations: Impact of pronunciation variations," in *Proceedings of InterSpeech*, 2014.

[15] R. Dall, S. Brognaux, K. Richmond, C. Valentini-Botinhao, G. Henter, J. Hirschberg, Y. J., and S. King, "Testing the consistency assumption: Pronunciation variant forced alignment in read and spontaneous speech synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, p. 51555159.

[16] K. Vythelingum, Y. Estève, and O. Rosec, "Error detection of grapheme-to-phoneme conversion in text-to-speech synthesis using speech signal and lexical context," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop*, 2017.

[17] J. R. Novak, P. R. Dixon, N. Minematsu, K. Hirose, C. Hori, and H. Kashioka, "Improving wfst-based g2p conversion with alignment constraints and rnnlm n-best rescoring," in *Proceedings of InterSpeech*, 2012.

[18] J. R. Novak, N. Minematu, and K. Hirose, "Failure transitions for joint n-gram models and g2p conversion," in *Proceedings of InterSpeech*, 2013.

[19] A. Stolcke, "Srilm – an extensible language modeling toolkit," in *Proceedings of InterSpeech*, 2002.

[20] A. Stolcke, J. Zheng, and W. Wang, "Srilm at sixteen: Update and outlook," in *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate." in *International Conference on Learning Representations (ICLR)*, 2015.

[22] O. Caglayan, M. García-Martínez, A. Bardet, W. Aransa, F. Bougares, and L. Barrault, "Nmtpy: A flexible toolkit for advanced neural machine translation systems," *Prague Bull. Math. Linguistics*, vol. 109, pp. 15–28, 2017. [Online]. Available: https://ufal.mff.cuni.cz/pbml/109/art-caglayan-et-al.pdf

[23] D. Povey, A. Ghoshal, G. Boulianne, L. Burge, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011.

[24] V. Peddinti, Y. Wang, D. Povey, and S. Khudanpur, "Low latency acoustic modeling using temporal convolution and lstms," *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 373–377, 2018.

[25] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Y. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Y. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu, "Deep speech 2: End-to-end speech recognition in english and mandarin," *CoRR*, vol. abs/1512.02595, 2015. [Online]. Available: http://arxiv.org/abs/1512.02595

[26] A. Graves, S. Fernndez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine learning*, 2006, p. 369376.