INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

On the Choice of Modeling Unit for Sequence-to-Sequence Speech Recognition

Kazuki Irie[1*], *Rohit Prabhavalkar*[2], *Anjuli Kannan*[2], *Antoine Bruguier*[2],
David Rybach[2], *Patrick Nguyen*[2]

[1]Human Language Technology and Pattern Recognition Group, Computer Science Department
RWTH Aachen University, D-52056 Aachen, Germany
[2]Google, Mountain View, CA 94043, USA

irie@cs.rwth-aachen.de, {prabhavalkar, anjuli, tonybruguier, rybach, drpng}@google.com

Abstract

In conventional speech recognition, phoneme-based models outperform grapheme-based models for non-phonetic languages such as English. The performance gap between the two typically reduces as the amount of training data is increased. In this work, we examine the impact of the choice of modeling unit for attention-based encoder-decoder models. We conduct experiments on the LibriSpeech 100hr, 460hr, and 960hr tasks, using various target units (phoneme, grapheme, and word-piece); across all tasks, we find that grapheme or word-piece models consistently outperform phoneme-based models, even though they are evaluated without a lexicon or an external language model. We also investigate model complementarity: we find that we can improve WERs by up to 9% relative by rescoring N-best lists generated from a strong word-piece based baseline with either the phoneme or the grapheme model. Rescoring an N-best list generated by the phonemic system, however, provides limited improvements. Further analysis shows that the word-piece-based models produce more diverse N-best hypotheses, and thus lower oracle WERs, than phonemic models.

Index Terms: End-to-end speech recognition, word-pieces, graphemes, phonemes, sequence-to-sequence

1. Introduction

Sequence-to-sequence learning [2] based on encoder-decoder attention models [3] has become popular for both machine translation [4] and speech recognition [5, 6, 7, 8, 9]. Such models are typically trained to output *character-based* units: graphemes, byte-pair encodings (BPEs) [10], or word-pieces [11], which allow the model to directly map the frame-level input audio features to the output word sequence, without using a hand-crafted pronunciation lexicon. Thus, when using such character-based output units, end-to-end speech recognition models [12] jointly learn the acoustic model, pronunciation model, and language model within a single neural network. In fact, such models outperform conventional hybrid recognizers [13] when trained on sufficiently large amounts of data [9].

One of the main advantages of character-based sequence-to-sequence models lies in their simplicity: both for training, as well as decoding. In fact, the use of characters as units for acoustic modeling has a long history for conventional HMM-based automatic speech recognition (ASR) systems (e.g., [14, 15, 16], inter alia). In the context of conventional ASR systems, for non-phonetic languages such as English, where the correspondence between orthography and pronunciation is less clear, previous works [14, 15] have found that phoneme-based models outperform grapheme-based models; grapheme-based systems approach the performance of phoneme-based systems only when much larger amounts of training training data are available [16]. It is therefore, natural to ask whether similar observations also apply to recently proposed attention-based encoder-decoder models: specifically, how do attention-based encoder-decoder models perform when using phonemes instead of character-based output units? To the best of our knowledge, this question has only been empirically investigated in the setting where a large amount of labeled training data are available. In previous work [17, 18], it has been empirically shown that the grapheme-based encoder-decoder models outperform the phoneme-based approach, while [17] find that use of lexica is still useful for recognizing rare words such as named entities.

In this work, we first investigate whether the previous result [17] which establishes the dominance of lexicon-free graphemic models over the phoneme-based models also hold on tasks with smaller amounts of training data. We carry out evaluations on the three subsets of the LibriSpeech task [19]: 100hr, 460hr, and 960hr, where we find that grapheme or word-piece models do indeed consistently outperform phoneme-based models, even when training data is limited. In Sec. 6, we further investigate the benefits offered by phonemic models by studying the complementarity of different units. In experimental evaluations, we find that simple N-best list rescoring results in large improvements in WER. Finally, we conduct a detailed analysis of the differences in the hypotheses produced by the models with various output units, in terms of quality of the top hypotheses, as well as the oracle error rate of the N-best list.

2. Sequence-to-Sequence Speech Models

All our models are Listen, Attend, and Spell (LAS) [12] speech models. The LAS model, which is depicted in Figure 1, has encoder, attention, and decoder modules. The *encoder* transforms the input frame-level audio feature sequence into a sequence of hidden activations. The *attention module* summarizes the encoder sequence into a single vector for each prediction step,

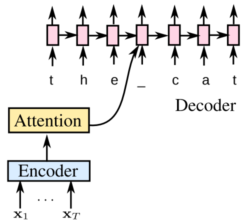

Figure 1: *LAS model.*

[*]Work performed during internship at Google. An initial version of this paper appears as a pre-print [1].

http://dx.doi.org/10.21437/Interspeech.2019-2277

and finally, the *decoder* models the distribution of the output sequence conditioned on the history of previously predicted labels. Both the encoder and the decoder are modeled using recurrent neural networks, and thus the entire model can be jointly optimized. We refer the interested reader to [12, 5, 20] for more details. Standard LAS models use *character-based* output units: grapheme [12], word-piece [9] or BPE [21].

3. Phonemic Sequence-to-Sequence Model

The phonemic LAS model can be obtained by using phonemes as the output unit. Phonemes are natural labels for acoustic modeling of non-phonetic languages. The use of a pronunciation lexicon can also ease integration of completely new words or named entities [22]. However, by using a pronunciation lexicon, we give up the end-to-end approach, which introduces complications for both training and decoding.

For training, words with multiple pronunciation variants cause a problem, since there is no unique mapping from such a word to its corresponding phoneme sequence. While we can potentially obtain the correct pronunciation variant by generating alignments, we skip this extra effort by choosing a pronunciation simply by randomly choosing one of the pronunciations for each word to define a unique mapping. In addition, we include an unknown token `UNK` as a part of the phoneme vocabulary and use it to represent words which are not included in the lexicon. We use a dedicated end-of-word token `EOW` (as part of the phoneme inventory) to model word boundaries, as in [17], which we find improves performance.

To deal with the ambiguity of homophones[1] during decoding, we incorporate a (word-based) n-gram language model. We use a general weighted finite-state transducer (WFST) decoder to perform a beam search. The lexicon and language model (LM) are represented as WFST L and G respectively and combined by means of FST composition as the search network $L \circ G$ [24]. The search process then explores partial path hypotheses which are constrained by the search network and scored by both the LAS model and the n-gram language model.

4. LibriSpeech Experimental Setup

4.1. Dataset

The LibriSpeech task [19] has three subsets with different amounts of transcribed training data: 100hr, 460hr, and 960hr. A lexicon with pronunciations for 200K words is officially distributed. The development and test data are both split into *clean* and *other* subsets, each of them consisting of about 5 to 6 hours of audio. The number of unique words observed in each subset as well as the out-of-vocabulary (OOV; unseen in training data) rate is summarized in Table 1. For language modeling, extra text-only data of about 800M words is also available, along with an officially distributed 3-gram word LM; we use the unpruned 3-gram LM for decoding the phonemic LAS models. In contrast, the grapheme and word-piece models are evaluated without a lexicon or a language model (unless otherwise indicated). We train word piece models [11] of size 16K (16,384) on each training subset.

Table 1: *Out-of-vocabulary (OOV) rates (%) with respect to the vocabulary (unique word list in the training data) in different data scenarios, and with respect to the pronunciation lexicon.*

Training data (h)	Vocab. Size	dev		test	
		clean	other	clean	other
100	34 K	2.5	2.5	2.4	2.8
460	66 K	0.9	1.2	1.0	1.3
960	89 K	0.6	0.8	0.6	0.8
Lexicon	200 K	0.3	0.6	0.4	0.5

4.2. Models and training

We use 80-dimensional log-mel features with deltas and accelerations as the frame-level audio input features. Reducing input frame rate in the encoder is important for successfully training sequence-to-sequence speech models, especially for tasks such as LibriSpeech which feature long utterances ($\sim$15s). Thus, following [25], our encoder layers include two layers of 3×3 convolution with 32 channels with a stride of 2, which results in a total time reduction factor of 4. We consider three model (*small*, *medium*, and *large*) which differ in terms of the sizes of model components. On top of the convolutional layers, the encoder contains 3 (*small*) or 4 (*medium* and *large*) layers of bidirectional LSTMs [26], with either 256 (*small*), 512 (*medium*), or 1024 (*large*) LSTM [27] cells in each layer. A projection layer and batch normalization are applied after each LSTM encoder layer [25]. The decoder consists of 1 (*small*) or 2 (*medium* and *large*) LSTM layers, and uses *additive attention* as described in [20].

We train all models using 16 GPUs by asynchronous stochastic gradient descent with Adam optimizer [28] from random initialization without any special pre-training method[2] for about 80 epochs. We use open-source Tensorflow Lingvo toolkit [29] for all experiments. Our grapheme and word-piece based baseline configurations are publicly available online[3] where further details about the models can be found.

5. Standalone Performance Results

5.1. Baseline model performance on 960hr

The WER performance of grapheme and word-piece based models is summarized in Table 2. For both graphemes and word-pieces, we present the performance for *small*, *medium* and *large* model sizes (as shown by different numbers of parameters) as described in Sec 4.2. The difference of number of parameters between different units only comes from the unit-level vocabulary size. As can be seen in Table 2, models benefit from the additional parameters and the best WERs are obtained for the large word-piece model.

Table 2: *WERs (%) for grapheme and word-piece models.*

Unit	Param.	dev		test	
		clean	other	clean	other
Grapheme	7 M	7.6	20.5	7.9	21.3
	35 M	5.3	15.6	5.6	15.8
	130 M	5.3	15.2	5.5	15.3
Word-Piece	20 M	5.8	16.0	6.1	16.4
	60 M	4.9	14.0	5.0	14.1
	180 M	**4.4**	**13.2**	**4.7**	**13.4**

[1] By choosing phonemes as output units, we are giving up the standalone recognition using the attention-based model. Also, while acoustic modeling motivates the use of phonemes, the ability of the decoder as a language model can possibly be weaker compared with character-based units, since the phoneme-level language model can be viewed as a subword-level class-based language model [23] where the clusters are formed based on the phonemic similarity.

[2] We find training to be stable across repeated runs. The initial learning rate is tuned to avoid plateaus at the start of training. Our models achieve the best WER on the dev-clean, earlier than on the dev-other.

[3] `https://github.com/tensorflow/lingvo`

5.2. Phonemic model performance on 960hr

For phoneme based models, we first check the phoneme error rates (PER) in order to make sure that the models are reasonable[4]. The WER performance results for decoding with the lexicon and the 3-gram word LM (88M n-grams) is shown in Table 3. We observe that despite the use of an external LM which is trained on much more data than the transcribed acoustic training data, the phonemic system performs worse than the best graphemic model[5]. It is nevertheless interesting to examine examples where the phonemic model outperforms the best word-piece model. In Table 4, we present some illustrative examples. In addition, we find that decoding the graphemic model with the 3-gram word LM does not give improvement.

Table 3: *WERs (%) for the* ***960hr*** *dataset.*

Unit	LM	dev		test	
		clean	other	clean	other
Phoneme	3-gram	5.6	15.8	6.2	15.8
Grapheme	None	5.3	15.2	5.5	15.3
Word-Piece 16K	None	4.4	13.2	4.7	13.4
Word-Piece 16K	LSTM	**3.3**	**10.3**	**3.6**	**10.3**
BPE 10K [21]	None	4.9	14.4	4.9	15.4
	LSTM	3.5	11.5	3.8	12.8
Hybrid system [30]	N-gram	3.4	8.8	3.6	8.9
	LSTM	3.1	8.3	3.5	8.6

Table 4: *Examples where the phonemic system's 1-best* ***wins*** *against the word-piece model's 1-best.*

Phoneme	Word-Piece
when did you come **bartley**	when did you come partly
kirkland jumped for the jetty	kerklin jumped for the jetty
man's eyes **remained** fixed	man's eyes were made fixed

In Table 3, we also include the WERs from previous work on LibriSpeech 960hr; for fair comparison, systems which employ data augmentation [31] are excluded. Our word-piece model performs better than the previously reported sequence-to-sequence model in [21] while the performance is behind the conventional hybrid system with an n-gram LM [30]. We note that our word-piece models simply trained using the cross-entropy criterion (without e.g. minimum word error rate training [32]) is competitive with Sabour et al.'s model trained with optimal completion distillation [33], which is reported to give 4.5% and 13.3% on the test-clean and test-other sets. For further comparison, we also report the WERs of our best word-piece model combined with an LSTM language model [34] by shallow fusion [35, 36][6]. The LSTM LM consists of one input linear layer of dimension 1024 and 2 LSTM layers with 2048 nodes [27]. The LM weight of 0.35 is found to be optimal for dev-clean and dev-other WERs. We obtain similar relative improvements reported in [21] and achieve WERs of 3.6% on the test-clean, and 10.3% on the test-other set, which reduces the performance gap from the best hybrid system reported in [30].

5.3. Results on 100hr and 460hr tasks

We conduct the same experiments in the 100hr and 460hr cases. For each unit, we obtain the best performance with the *large* model for the 460hr scenario, and the *medium* model for the 100hr case. The results are summarized in Table 5. We find that even in the small dataset scenarios with higher OOV rates, graphemic and word-piece based models outperform the phonemic system. We also note that the performance of attention-based models dramatically degrades when the amount of training data is reduced, unlike conventional hybrid approach [19].

Table 5: *WERs for the* ***460hr*** *and* ***100hr*** *scenarios.*

Train data	Unit	dev		test	
		clean	other	clean	other
460hr	Phoneme	7.6	27.3	8.5	27.8
	Grapheme	6.4	23.5	6.8	24.1
	Word-Piece	**5.7**	**21.8**	**6.5**	**22.5**
100hr	Phoneme	13.8	38.9	14.3	40.9
	Grapheme	**11.6**	36.1	**12.0**	38.0
	Word-Piece	12.7	**33.9**	12.9	**35.5**

6. Rescoring Experiments

While Sec. 5 focuses on the comparison of models with different output units, our goal is to ideally get benefits from different model units. We consider two methods for combining LAS models with different units. The first approach is simple *N-best list rescoring*: generate a N-best list from one LAS model, convert the corresponding word sequences to the rescorer LAS model's unit, and combine the scores by log-linear interpolation. However, rescoring is limited to the hypotheses generated by one LAS model. Therefore, we also carry out *union of N-best list with cross-rescoring*: we independently generate N-best lists from two LAS models, rescore the hypotheses generated by one model using the other model and vice versa, to get the 1-best from the union of the rescored (up to) 2N hypotheses.

6.1. N-best rescoring results

We carry out the N-best rescoring of our best word-piece based model in the 960hr scenario by a graphemic, and a phonemic model. The interpolation weights are optimized to obtain the best dev-clean WER (which typically also gives the best dev-other WER). The WERs are presented in the upper part of Table 6. We obtain improvements of 9% in both cases on the test-clean set; on the test-other set, we obtain an 8% relative with the phonemic model and a 9% relative with the graphemic model. Thus, it can be noted that rescoring is a simple method for using a phonemic model without an additional language model. To determine if gains by the graphemic and phonemic models are additive, we combine the scores from all models, which obtains only slight improvements of up to 0.1 absolute as shown in Table 6 (+ Both). In Table 7, we again show some illustrative examples where the phonemic model outperforms the combination of word-piece and grapheme based models only. It is for example interesting to observe that the correct spelling "bartley" is in the N-best hypotheses of the word-piece model, and that the phonemic model helps recognize it correctly.

In the other direction, we also rescore the N-best list generated by the phonemic system by the word-piece model. The results are shown in the lower part of Table 6. We find that the improvements are limited (only up to 4% relative). In fact, the 30-best list generated by a phonemic system has much higher oracle WERs than the 8-best list of the word-piece model.

[4] By increasing the model size from 7 M to 35 M, then to 130 M, we improve the PERs (%) from (3.2, 9.7, 3.2, 9.9), to (2.8, 8.9, 3.0, 9.1), then to (2.4, 7.9, 2.5, 7.7) on the dev-clean/other, test-clean/other sets.

[5] As also reported in [17]; though, we note that we get about 2% absolute degradation in WERs with a model trained without EOW.

[6] We find it crucial to constrain the emission of end-of-sentence (EOS) tokens [35] to penalize short sentences (rather than applying length normalization) in shallow fusion: we only allow the model to emit EOS when its score is within 1.0 of the top hypothesis. We check that such an EOS penalty does not improve the baseline systems without LM nor beam search with the WFST decoder for phonemic models.

Table 6: *WER (%) results for N-best list rescoring. Oracle WERs are shown in parentheses.*

	dev		test	
	clean	other	clean	other
Word-Piece	4.4 (2.4)	13.2 (9.2)	4.7 (2.6)	13.4 (9.1)
+ Phoneme	4.1	12.4	**4.3**	12.4
+ Grapheme	4.0	12.3	**4.3**	12.3
+ Both	**3.9**	**12.2**	**4.3**	**12.2**
Phoneme	5.6 (4.9)	15.8 (14.4)	6.2 (5.5)	15.8 (14.7)
+ Word-Piece	5.4	15.5	6.0	15.5

Table 7: *Examples where Word-Piece+Grapheme+Phoneme (WP+G+P)* ***wins*** *over Word-Piece+Grapheme (WP+G).*

WP+G+**P**	WP+G
oh **bartley** did you write to me	oh bartly did you write to me
... lettuce leaf with **mayonnaise** ...	... lettuce leaf with mayonna is ...
the manager **fell to** his musings	the manager felt of his musings
what **a fuss** is made about you	what are fusses made about you
... eyes **blazed with** indignation	... eyes blaze of indignation

6.2. Union of N-best lists with cross-rescoring results

The examples in Table 4 show some complementarity between the word-piece 1-best hypothesis and the phonemic one. To evaluate the potential value of hypotheses generated by the phonemic model, we decode a N-best list from the word-piece and phoneme models independently, rescore the respective hypotheses (cross-rescoring), and take the 1-best from the 2N hypotheses (union). Table 8 shows only marginal improvements on the test-other set, compared with rescoring the 8-best word-piece hypotheses. For a fairer comparison, we also carry out rescoring of 16-best lists generated by the word-piece model by the phonemic model. We find that such an approach is slightly better than the union. This suggests that decoding from the phonemic model has limited benefits for the LibriSpeech task.

Table 8: *WERs (%) results for union of N-best lists with cross-rescoring. Oracle WERs are shown in parentheses.*

	Num.	dev		test	
	hyp.	clean	other	clean	other
Word-Piece	8	4.4 (2.4)	13.2 (9.2)	4.7 (2.6)	13.4 (9.1)
+ Phoneme		4.1	12.4	4.3	12.4
Union	16	4.1	12.4	4.3	12.3
Word-Piece	16	4.4 (2.0)	13.2 (8.3)	4.7 (2.2)	13.4 (8.1)
+ Phoneme		**4.0**	**12.3**	**4.3**	**12.2**

6.3. Why is oracle WER so high for phonemic system?

The oracle WERs are much worse for the phonemic system than the word-piece model (Table 6). We observe that the diversity of hypotheses in the N-best list generated by the phonemic system is mainly based on homophones, rather than *difficult* words (i.e. words with unusual pronunciation). For example, on the reference utterance *"bozzle had always waited upon him with a decent coat and a well brushed hat and clean shoes"*, where *bozzle* is not in the training data, the word-piece based model fills the 8-best beam by proposing different spellings for *bozzle* such as {basil, bazil, basle, bosel, bosal, bosell, bossel}, which is a reasonable way to model the ambiguity. The phoneme system, instead, only produces {bazil, basil} as a substitution for *bozzle* and lists homophones for *shoes*, {shoes, shews, shoos, shues, shooes} instead. Homophone distinction might still be inefficient for a phonemic system as the phonemic LAS model gives them all the same score, and a single parameter is used to weight the external LM for the entire search. Addressing this issue might be crucial to improve the phonemic system.

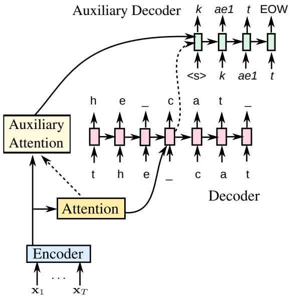

Figure 2: *LAS model with an auxiliary decoder. Dashed lines represent state copying for initialization at each word boundary.*

6.4. Rescoring with an auxiliary decoder

Finally, we examine a model with two decoders operating on different units but using a single encoder. Such a model can be convenient for model combination (e.g., rescoring or potentially also decoding from two decoders operating on different units and combining hypotheses in a word synchronous fashion). The design of the model is illustrated in Figure 2. The main decoder (grapheme, in the example) works exactly as in the baseline LAS model (Sec. 2; Figure 1). The auxiliary decoder (phoneme, in the example) is designed such that it predicts *only the next word as a sequence of the auxiliary units*. We use separate parameters for the auxiliary attention and initialize all recurrent states of the auxiliary component at each word boundary by those of the main decoder (i.e. the prediction from the auxiliary decoder is conditioned on the word sequence generated thus far from the main decoder). The model is trained in two stages; the main decoder and the encoder are first trained, and their parameters are not modified during the training of the auxiliary components. In experiments, we use word-pieces for the main decoder, and phonemes for the auxiliary decoder. Table 9 shows improvements by rescoring with an auxiliary phoneme decoder of the two decoder-model. We obtain improvements despite small number of additional parameters (30M) corresponding to the phonemic 2-layer LSTM decoder and the attention layer, however rescoring with an independent phoneme model (as in Table 6) gives larger improvements.

Table 9: *WERs (%) for rescoring with an auxiliary decoder.*

	dev		test		Total
	clean	other	clean	other	Param.
Word-Piece (WP)	4.4	13.2	4.7	13.4	180 M
WP + Auxiliary phoneme	4.3	13.0	4.6	13.1	210 M
WP + Phoneme	4.1	12.4	4.3	12.4	310 M

7. Conclusion

Our experiments on different LibriSpeech subsets show that word-piece and grapheme models consistently outperform phoneme models. Therefore, the dominance of character-based model units in the LAS speech model is not due to the amount of training data. This indicates that this behavior is more likely related to the model itself (e.g., the decoder is conditioned on all predecessor labels). Furthermore, we find that the word-piece based attention models can achieve a relatively low oracle WER with only 8-best hypotheses and rescoring that N-best hypotheses using graphemic or phonemic models gives good improvements. Future work will examine whether streaming end-to-end approaches (e.g., RNN-T [37, 38]) show similar trends.

8. Acknowledgements

We thank Tara Sainath and Yu Zhang for helpful discussion, and Jinxi Guo for sharing his language model setup.

9. References

[1] K. Irie, R. Prabhavalkar, A. Kannan, A. Bruguier, D. Rybach, and P. Nguyen, "Model unit exploration for sequence-to-sequence speech recognition," *preprint arXiv:1902.01955*, 2019.

[2] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, Montreal, Canada, Dec. 2014, pp. 3104–3112.

[3] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.

[4] Y. Wu, M. Schuster, Z. Chen *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[5] R. Prabhavalkar, K. Rao, T. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *Proc. Interspeech*, Stockholm, Sweden, Aug. 2017, pp. 939–943.

[6] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 4835–4839.

[7] E. Battenberg, J. Chen, R. Child, A. Coates, Y. G. Y. Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu, "Exploring neural transducers for end-to-end speech recognition," in *Proc. ASRU*, Okinawa, Japan, Dec. 2017, pp. 206–213.

[8] C. Weng, J. Cui, G. Wang, J. Wang, C. Yu, D. Su, and D. Yu, "Improving attention based sequence-to-sequence models for end-to-end english conversational speech recognition," in *Proc. Interspeech*, Hyderabad, India, Sep. 2018, pp. 761–765.

[9] C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. ICASSP*, Calgary, Canada, Apr. 2018, pp. 4774–4778.

[10] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *ACL*, Berlin, Germany, August 2016, pp. 1715–1725.

[11] M. Schuster and K. Nakajima, "Japanese and korean voice search," in *Proc. ICASSP*, Kyoto, Japan, Mar. 2012, pp. 5149–5152.

[12] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: a neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, Shanghai, China, Mar. 2016, pp. 4960–4964.

[13] H. A. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*. Norwell, MA, USA: Kluwer Academic Publishers, 1993.

[14] S. Kanthak and H. Ney, "Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Orlando, FL, USA, May 2002, pp. 845–848.

[15] M. Killer, S. Stüker, and T. Schultz, "Grapheme based speech recognition," in *Proc. Eurospeech*, Geneva, Switzerland, Sep. 2003.

[16] Y. Sung, T. Hughes, F. Beaufays, and B. Strope, "Revisiting graphemes with increasing amounts of data," in *Proc. ICASSP*, Taipei, Taiwan, Apr. 2009, pp. 4449–4452.

[17] T. N. Sainath, R. Prabhavalkar, S. Kumar, S. Lee, A. Kannan, D. Rybach, V. Schogol, P. Nguyen, B. Li, and Y. Wu, "No need for a lexicon? Evaluating the value of the pronunciation lexica in end-to-end models," in *Proc. ICASSP*, Calgary, Canada, Apr. 2018, pp. 5859–5863.

[18] S. Zhou, L. Dong, S. Xu, and B. Xu, "A comparison of modeling units in sequence-to-sequence speech recognition with the Transformer on Mandarin Chinese," *arXiv preprint arXiv:1805.06239*, 2018.

[19] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: an ASR corpus based on public domain audio books," in *ICASSP*, South Brisbane, Queensland, Australia, Apr. 2015, pp. 5206–5210.

[20] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," in *Interspeech*, Stockholm, Sweden, Aug. 2017, pp. 2625–2629.

[21] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," in *Interspeech*, Hyderabad, India, Sep. 2018, pp. 7–11.

[22] A. Bruguier, R. Prabhavalkar, G. Pundak, and T. N. Sainath, "Phoebe: Pronunciation-aware contextualization for end-to-end speech recognition," in *Proc. ICASSP*, Brighton, England, May 2019.

[23] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[24] M. Mohri, F. Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," in *Handbook of Speech Processing*. Springer, 2008, ch. 28, pp. 559–582.

[25] Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 4845–4849.

[26] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, San Diego, CA, USA, May 2015.

[29] J. Shen, P. Nguyen, Y. Wu, Z. Chen *et al.*, "Lingvo: a modular and scalable framework for sequence-to-sequence modeling," *arXiv preprint:1902.08295*, 2019.

[30] K. J. Han, A. Chandrashekaran, J. Kim, and I. Lane, "The CAPIO 2017 conversational speech recognition system," *arXiv preprint:1801.00059*, 2018.

[31] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[32] R. Prabhavalkar, T. N. Sainath, Y. Wu, P. Nguyen, Z. Chen, C. Chiu, and A. Kannan, "Minimum word error rate training for attention-based sequence-to-sequence models," in *Proc. ICASSP*, Calgary, Canada, Apr. 2018, pp. 4839–4843.

[33] S. Sabour, W. Chan, and M. Norouzi, "Optimal completion distillation for sequence learning," in *Int. Conf. on Learning Representations (ICLR)*, New Orleans, LA, USA, May 2019.

[34] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling." in *Proc. Interspeech*, Portland, OR, USA, Sep. 2012, pp. 194–197.

[35] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *Internspeech*, Stockholm, Sweden, Aug. 2017, pp. 523–527.

[36] S. Toshniwal, A. Kannan, C.-C. Chiu, Y. Wu, T. N. Sainath, and K. Livescu, "A comparison of techniques for language model integration in encoder-decoder speech recognition," in *Proc. SLT*, Athens, Greece, Dec. 2018.

[37] A. Graves, "Sequence transduction with recurrent neural networks," in *Representation Learning Workshop, Int. Conf. on Machine Learning (ICML)*, Edinburgh, Scotland, Jun. 2012.

[38] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer," in *Proc. ASRU*, Okinawa, Japan, Dec. 2017, pp. 193–199.