A SYSTEMATIC COMPARISON OF GRAPHEME-BASED VS. PHONEME-BASED LABEL UNITS FOR ENCODER-DECODER-ATTENTION MODELS

Mohammad Zeineldeen, Albert Zeyer, Wei Zhou, Thomas Ng, Ralf Schlüter, Hermann Ney

Human Language Technology and Pattern Recognition, Computer Science Department,
RWTH Aachen University, 52062 Aachen, Germany,
AppTek GmbH, 52062 Aachen, Germany

{zeineldeen, zeyer, zhou, schlueter, ney}@cs.rwth-aachen.de, thomas.ng@rwth-aachen.de

ABSTRACT

Following the rationale of end-to-end modeling, CTC, RNN-T or encoder-decoder-attention models for automatic speech recognition (ASR) use graphemes or grapheme-based subword units based on e.g. byte-pair encoding (BPE). The mapping from pronunciation to spelling is learned completely from data. In contrast to this, classical approaches to ASR employ secondary knowledge sources in the form of phoneme lists to define phonetic output labels and pronunciation lexica. In this work, we do a systematic comparison between grapheme- and phoneme-based output labels for an encoder-decoder-attention ASR model. We investigate the use of single phonemes as well as BPE-based phoneme groups as output labels of our model. To preserve a simplified and efficient decoder design, we also extend the phoneme set by auxiliary units to be able to distinguish homophones. Experiments performed on the Switchboard 300h and LibriSpeech benchmarks show that phoneme-based modeling is competitive to grapheme-based encoder-decoder-attention modeling.

Index Terms— end-to-end speech recognition, attention, phonemes, byte pair encoding

1. INTRODUCTION & RELATED WORK

End-to-end models such as encoder-decoder-attention models [1–6], connectionist temporal classification (CTC) models [7–10], or the recurrent neural network transducer (RNN-T) [11, 12] have shown competitive performance for speech recognition. End-to-end models usually operate on

- *graphemes* (characters),
- *graph. based subword units* (*byte-pair encoding (BPE)* [3, 13, 14], word piece model (WPM) [15, 16], unigram language model (LM) based segmentation [17, 18], latent sequence decompositions [19, 20], or pronunciation-assisted char. subwords [21])
- or *whole words* [22–27].

Grapheme-based modeling bears the advantages of easily allowing to recognize out-of-vocabulary words, enabling modeling, training and decoding simplicity and avoiding potential errors introduced by secondary knowledge sources like pronunciation lexica. The disadvantage of grapheme-based modeling might be limited scalability to low-resource situations. Phoneme-based modeling on the other hand easily enables adaptation to new pronunciations or inclusion of words with uncommon pronunciations [28]. These output-label modeling alternatives therefore provide a tradeoff between simplicity, flexibility, and controllability.

Hybrid neural network (NN) - hidden Markov models (HMMs) [29, 30] usually operate on context-dependent phonemes [31, 32], although it has been shown that they can also work on context-dependent graphemes [33–36]. Due to the resulting combinatorial complexity, phonemes (graphemes) in context usually are clustered via classification and regression trees (CART) [37, 38].

All end-to-end models easily allow for phoneme-based labels as well, although in most cases that requires a more complex decoder like a weighted finite-state transducer (WFST) decoder [39], and a pronunciation lexicon. Phoneme-based CTC models (context-independent or clustered context-dependent) [25, 40, 41] usually perform better than grapheme-based CTC models, just as hybrid NN-HMMs. Phoneme-based encoder-decoder-attention models [39, 42–45] have shown mixed results so far – in most cases the performance was slightly worse than grapheme-based attention models, or only the combination of both helped. More recently also phoneme-based RNN-T-like models [46, 47] were studied, although it is unclear whether pure phoneme-based RNN-T models perform better than pure grapheme-based models.

Subwords based on phonemes, like phoneme-BPE, was only studied so far by [44], which is a hybrid CTC / attention model [48]. They use multiple LMs in decoding: a phoneme-BPE LM and a word LM.

In this work, we perform a systematic comparison of grapheme-based and phoneme-based output labels with an encoder-decoder-attention model.

2. VARIATION OF LABEL UNITS

We want to perform a systematic study of the difference between phonemes, graphemes, subwords based on phonemes or graphemes, and whole words. In this work, we focus on

attention-based encoder-decoder models. We note that the results of such label unit study will likely look different depending on the type of model. E.g. subwords and words can possibly have variations in their observed audio lengths, which might be less of a problem for label-synchronous models but more a problem for time-synchronous models. For hybrid NN-HMMs or CTC models, we also know that clustered context-dependent labels help a lot [36, 38], while models with label feedback such as RNN-T and encoder-decoder already cover the (left) context. Hybrid NN-HMMs also usually split the phonemes into begin-middle-end states. As we perform the experiments with an encoder-decoder-attention model which is label-synchronous and auto-regressive, we also have the end-of-sentence (EOS) token in our vocabulary. In case of a time-synchronous model, we possibly would add a blank label (like for CTC or RNN-T), or maybe repetition symbols (as in the Auto Segmentation Criterion (ASG) [49]), or a silence label (for hybrid NN-HMMs). It also might make sense to add noise or unknown labels (e.g. as in [39]).

The label units which we are going to compare are:
- Phonemes
  - Single phonemes
  - Single phonemes + end-of-word (EOW) symbol
  - Single phonemes + word-end-phones (#)
  - Phoneme-BPE

  Variation: Extra word-disambiguate symbols
- Graphemes (characters)
  - Single characters + whitespace
  - Char-BPE
- Whole words

Phoneme-BPE is simply the application of BPE on phoneme sequences. We generate the phoneme BPE codes by taking a single pronunciation for each word (the most likely as defined by the lexicon) for all the transcriptions. The same phoneme sequence is also used for training. The end-of-word (EOW) symbol for phonemes is similar as white-space character for grapheme-based models. Others [39, 42] have observed that the EOW symbol can be helpful for phoneme models. We follow [50] to evaluate another variant for single phonemes where for each phoneme label x, we add another label with word-end-phone marker "#" yielding "x#". This will make the vocab. size twice larger.

For phoneme-based models, we need a lexicon for the mapping of phonemes to words. But then there are cases where a phoneme sequence can be mapped to multiple possible words (*homophones*). E.g. the word "I" and "eye" both have the same pronunciation consisting of the phoneme sequence "ay". To be able to discriminate between "I" and "eye", usually an external language model on word-level is used. Alternatively, we can also add special *word-disambiguate symbols* to the labels (the phoneme inventory), in such a way that we can always uniquely discriminate between all words. We go through the pronunciation lexicon

and collect all phoneme sequences which can not be uniquely mapped to words. For all these phoneme sequences, we add special symbols $1 to $N. For example "ay $8" → "eye", "ay $9" → "I", ..., "r eh d $2" → "read", "r eh d $3" → "red". These word-disambiguate symbols allow for decoding without an external language model, and also allows us to use our simplified decoder implementation. It also might improve the performance as the model has now the power to discriminate between words. Note that this scheme of adding these symbols does not allow for an easy extension of the lexicon for further homophones after we trained the model.

3. MODEL

We use an encoder-decoder-attention model [51, 52]. Our encoder consists of 6 layers of bidirectional long short-term memory (LSTM) [53] networks with time-downsampling via max-pooling by factor 6. Our decoder is a single layer LSTM network with global attention. We use SpecAugment [4] for on-the-fly data augmentation. For further details, please refer to our earlier work [3, 5], which is exactly the same model, except for the variations of the output label.

3.1. Training

We maximize the log-likelihood $\log p(y_1^N | x_1^T)$, for target sequences y_1^N and input feature sequences x_1^T over the training dataset. We always have a single ground-truth target sequence. In the case of phonemes, we reduce the lexicon to contain only a single pronunciation per word, and thus this becomes unique. This is a simplification, which we only do for training, not for decoding.

3.2. Decoding

Our *simplified decoder* performs beam search over the labels with a fixed beam size (e.g. 12 hypotheses) without any restrictions (i.e. it allows any possible label sequence). Our simplified decoder allows for log-linear combination with a LM on the same label-level (e.g. phone-BPE) but not with word-level LM. *After* we find a label sequence with this simplified beam search, we map it to words. In case of BPE, we first do BPE merging. In case of phonemes with word-disambiguate symbols, we try to lookup the corresponding word (which is unique because of the word-disambiguate symbols), or replace it by some UNK symbol if not found. That way, we eventually end up with a sequence of words.

Our *advanced decoder* performs prefix-tree search based on a lexicon. This lexicon defines the mapping between words and corresponding phoneme or grapheme label sequences. The resulting lexical tree restricts the search to possible label sequences from the lexicon. It also allows log-linear combination with a word-level LM. The LM score is applied to a hypothesized path whenever it reaches a word-end. Optionally, LM lookahead can also be applied to incorporate the LM score into the tree for a more robust search. The standard beam pruning using a fixed beam size is applied at each search step. Finally, the decoded best path directly gives the recognized word sequence.

4. EXPERIMENTS

We use RETURNN [54]. The advanced decoder is implemented as part of RASR [55], while the simplified decoder is implemented within RETURNN. All our config files and code to reproduce these experiments can be found online[1].

4.1. Switchboard 300h

We perform experiments on the Switchboard 300h English telephone speech corpus [56]. We use Hub5'00 as a development set, which consists of both Switchboard (SWB) and CallHome (CH), and Hub5'01 as a test set.

During training, because of GPU memory constraints, we filter out too long sequences based on a threshold on the target sequence length. The length of the target sequences varies depending on the labels unit, e.g. a sequence of phonemes can be much longer than with grapheme-BPEs. Thus, to have a fair comparison, we set the maximum allowed target sequence length for all label units individually such that we drop the same amount of training utterances (0.35% are dropped).

We collect our experiments with our simplified decoder without LM in Table 1. The simplified decoder can only produce reasonable results if the label units allow for word disambiguations, such as in the case of graphemes, but also for phonemes with added word-disambiguate symbols. We find that BPE subwords perform much better than single units and also better than whole words, both for phonemes and graphemes. In later experiments with the advanced decoder, where the search is restricted to valid sequences from the lexicon, we will see that single phoneme units perform much better. Moreover, word-end-phones perform better than EOW symbol, which is also different with the advanced decoder. For grapheme-BPE, BPE-534 model seems to perform best whereas for phoneme-BPE, both BPE-592 and BPE-1k models are comparable. Note that the num. of labels has an impact on the training time, due to the softmax output layer, and different output sequence lengths. Single (EOW) phoneme training is 10-20% relatively slower than BPEs or whole words.

We compare different decoding variants in Table 2. Our advanced decoder restricts the search to only label sequences which occur in the lexicon, including only the BPE-splits seen during training, in contrast to the simplified decoder, which does not have this restriction. We see that in the case of single phone or grapheme labels, i.e. where we have a weaker model, the restriction on the lexicon by the advanced decoder is significantly helpful (consistent with [10]) while it is hurtful for the BPE variants, esp. in the case of phoneme-BPE. We also see that the word-level LM improves the performance in all cases (as expected).

We study the effect of the word-disambiguate symbols for phoneme-based models in Table 3. We find that the word-disambiguate symbols seem to be hurtful. We are still careful in drawing conclusions from this, as this might be due to the specific variant of how we added such symbols.

Table 1: Comparing **phoneme**, **grapheme** and **whole word** models on Switchboard 300h. Using simplified decoding with beam size 12 without language model nor lexicon. All phoneme models have *word-disambiguate symbols*. Single phoneme is with end-of-word (EOW) symbol or word-end-phone symbol (#). Single grapheme is with whitespace.

Labels			WER[%]			
Unit	Type	#Num	Hub5'00			Hub5'01
			SWB	CH	Σ	Σ
Phoneme	Single (EOW)	62	12.6	30.7	21.7	18.2
	Single (#)	118	11.6	25.7	18.7	16.7
	BPE	151	10.9	23.9	17.4	16.6
		201	10.6	22.8	16.7	15.9
		592	9.7	20.8	**15.3**	14.9
		1k	10.0	20.4	**15.2**	15.1
		2k	10.3	21.4	15.9	15.4
		5k	10.5	21.6	16.0	15.5
Grapheme	Single	35	13.3	32.8	23.1	18.6
	BPE	126	10.3	22.9	16.6	15.8
		176	9.7	21.0	15.3	14.7
		534	10.0	20.6	**15.3**	14.8
		1k	10.3	21.1	15.7	15.5
		2k	10.0	21.4	15.7	14.7
		5k	10.4	21.8	16.1	15.5
		10k	10.9	22.7	16.8	15.9
		20k	11.4	22.9	17.2	16.6
Words	Single	30k	12.5	25.0	18.8	18.0

For single phoneme labels, we also study the effect of having an end-of-word (EOW) symbol or not, or using the word-end-phones variant. All the results are with the advanced decoder, i.e. restricted only to valid entries from the lexicon, and with LM. We show the results in Table 4. Consistent with [39], having EOW helps. In this setting, we observe that word-end-phones are worse, in contrast to the simplified decoder without restriction from the lexicon.

Finally, we compare our results to other results from the literature in Table 5. We observe that many other works train for much longer, and that seems to lead to yet better results. Our phoneme-based models perform slightly better than our final grapheme-based models, although they are very close.

4.2. LibriSpeech 960h

Furthermore, we conduct experiments on the LibriSpeech 960h corpus [59]. All models are trained for 13 epochs. Our results are in Table 6. We observe that grapheme and phoneme BPE outperform single phonemes, in contrast to our Switchboard results. This might be due to the 0.7% OOV rate. We also observed that specifically the single phoneme model overfits much more than all other experiments. Grapheme BPE-10k reports the best WER on all data sets.

5. CONCLUSIONS

We compared phoneme-based labels vs. grapheme-based labels for attention-based encoder-decoder models and found their performance to be similar – the phoneme-based models are slightly better on the smaller Switchboard corpora, but worse on the LibriSpeech corpora. We also compared single

Table 2: **Decoding comparison** on Switchboard 300h. All phoneme models here have word-disambiguate symbols. Single phoneme is with EOW symbol. Single grapheme is with whitespace. All with beam size 12 and the optional LSTM LM is on word-level. The advanced decoder is also restricted on the lexicon and the unique greedy BPE-split.

Labels		Decoder		WER[%]			
				Hub5'00			Hub5'01
Unit	Type	LM	Impl.	SWB	CH	Σ	Σ
Phon.	Single (EOW)	None	Simplified	12.6	30.7	21.7	18.2
			Advanced	9.9	21.2	15.6	15.2
		LSTM	Advanced	9.0	20.6	**14.8**	14.1
	BPE-592	None	Simplified	9.7	20.8	15.3	14.9
			Advanced	10.5	22.3	16.4	16.1
		LSTM	Advanced	9.5	21.6	15.6	14.8
Graph.	Single	None	Simplified	13.3	32.8	23.1	18.6
			Advanced	10.4	21.9	16.2	15.4
		LSTM	Advanced	9.3	21.1	15.3	14.2
	BPE-534	None	Simplified	10.0	20.6	15.3	14.8
			Advanced	10.1	20.5	15.3	15.0
		LSTM	Advanced	8.9	20.2	**14.6**	13.9

Table 3: Studying **word-disambiguate symbols** on Switchboard 300h by comparing different **phoneme** variants. All with word-level LSTM LM, and the advanced decoder.

Phoneme Labels			Beam size	WER[%]		
				Hub5'00		
Type	#Num	Disamb.		SWB	CH	Σ
Single (EOW)	62	Yes	12	9.0	20.6	14.8
			32	9.3	20.5	14.9
	48	No		**8.7**	**20.1**	**14.4**
BPE	592	Yes	12	9.5	21.6	15.6
			64	10.1	22.6	16.3
	588	No		8.7	20.7	14.7

units vs. subword (BPE) units vs. whole words, and found that subword units are best, both for phonemes and graphemes. For single units, the restriction to valid sequences via lexicon in search is crucial for good performance.

6. ACKNOWLEDGEMENTS

7. REFERENCES

[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015.

[2] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," Preprint arXiv:1508.04025, 2015.

[3] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," in *Interspeech*, 2018.

[4] 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," in *Interspeech*, 2019.

[5] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney, "A comparison of Transformer and LSTM encoder decoder models for ASR," in *ASRU*, 2019.

[6] Z. Tüske, G. Saon, K. Audhkhasi, and B. Kingsbury, "Single headed attention based sequence-to-sequence model for state-of-the-art results on switchboard-300," Preprint arXiv:2001.07263, 2020.

[7] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[8] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based de-

Table 4: Comparing **end-of-word (EOW)** and **word-end-phones** variant for single phonemes. WER is on Hub5'00. All results are with word-level LM and advanced decoder, beam size 32, without word disambiguate symbols.

Variant	#Num	WER[%]
No EOW	47	14.8
EOW	48	**14.4**
Word-end-phones (#)	90	15.5

Table 5: Comparing results from the **literature** on Switchboard 300h. One big difference in varying results is the different amount of training time (number of epochs). *no train seq. len. filter.

Work	Label			#Ep	LM	WER[%]				
						Hub5'00			Hub5'01	RT'03
	Unit	Type	#Num			SWB	CH	Σ	Σ	Σ
[32]	Phon.	CART	4.5k	13	Yes	9.6	18.5	14.0	14.1	
[5]	Graph.	BPE	1k	33	No	10.1	20.6	15.4	14.7	
					Yes	9.3	20.3	14.9	14.2	
[57]			4k	50	No	8.8	17.2	13.0		
[58]			2k	100	Yes	9.0	18.1	13.6		
[44]	Phon.		500	150	Yes	7.9	16.1	12.0		14.5
[6]	Graph.		600	250	No	7.6	14.6	11.1		
					Yes	6.4	12.5	9.5		
[4]		WPM	1k	760	No	7.2	14.6	10.9		
					Yes	6.8	14.1	10.5		
Ours	Graph	BPE	534	33	No	10.1	20.5	15.3	15.0	18.0
					Yes	8.9	20.2	14.6	13.9	16.9
	Phon.	Single	47			**8.7**	20.1	14.4	**13.8**	16.7
		BPE	588			**8.7**	20.7	14.7	**13.8**	17.5
Ours*						**8.7**	**19.7**	**14.2**	**13.8**	**16.2**

Table 6: Comparing models with different label units on LibriSpeech 960h. All phoneme models are without word-disambiguate symbols. All with word-level LSTM LM, and the advanced decoder.

Labels			WER[%]			
			dev		test	
Unit	Type	#Num	clean	other	clean	other
Phoneme	Single (#)	141	5.14	12.83	5.71	13.79
	BPE	5k	3.42	8.91	3.91	9.74
		10k	3.28	8.82	3.88	9.31
		20k	3.50	9.06	3.86	10.56
Grapheme	BPE	5k	3.35	8.41	3.84	9.48
		10k	**3.15**	**8.32**	**3.59**	**9.14**
		20k	3.46	9.35	3.93	10.08

coding," in *ASRU*. IEEE, 2015.

[9] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," Preprint arXiv:1512.02595, 2015.

[10] G. Zweig, C. Yu, J. Droppo, and A. Stolcke, "Advances in all-neural speech recognition," in *ICASSP*, 2017.

[11] A. Graves, "Sequence transduction with recurrent neural networks," Preprint arXiv:1211.3711, 2012.

[12] A. Zeyer, A. Merboldt, R. Schlüter, and H. Ney, "A new training pipeline for an improved neural transducer," in *Interspeech*, 2020.

[13] P. Gage, "A new algorithm for data compression," *C Users Journal*, vol. 12, no. 2, 1994.

[14] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," Preprint arXiv:1508.07909, 2015.

[15] M. Schuster and K. Nakajima, "Japanese and korean voice search," in *ICASSP*, 2012.

[16] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," Preprint arXiv:1609.08144, 2016.

[17] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *ACL*, 2018.

[18] K. Bostrom and G. Durrett, "Byte pair encoding is suboptimal for language model pretraining," Preprint arXiv:2004.03720, 2020.

[19] W. Chan, Y. Zhang, Q. Le, and N. Jaitly, "Latent sequence decompositions," Preprint arXiv:1610.03035, 2016.

[20] J. Drexler and J. Glass, "Learning a subword inventory jointly with end-to-end automatic speech recognition," in *ICASSP*, 2020.

[21] H. Xu, S. Ding, and S. Watanabe, "Improving end-to-end speech recognition with pronunciation-assisted sub-word modeling," in *ICASSP*, 2019.

[22] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," in *Interspeech*, 2017.

[23] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, "Direct acoustics-to-word models for english conversational speech recognition," in *Interspeech*, 2017.

[24] S. Palaskar and F. Metze, "Acoustic-to-word recognition with sequence-to-sequence models," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018.

[25] Z. Chen, Q. Liu, H. Li, and K. Yu, "On modular training of neural acoustics-to-word model for LVCSR," in *ICASSP*, 2018.

[26] K. Audhkhasi, B. Kingsbury, B. Ramabhadran, G. Saon, and M. Picheny, "Building competitive direct acoustics-to-word models for english conversational speech recognition," in *ICASSP*, 2018.

[27] A. Das, J. Li, G. Ye, R. Zhao, and Y. Gong, "Advancing acoustic-to-word CTC model with attention and mixed-units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, 2019.

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

[29] H. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*, Springer, 1994.

[30] A. J. Robinson, "An application of recurrent nets to phone probability estimation," *Neural Networks, IEEE Transactions on*, vol. 5, no. 2, 1994.

[31] A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schlüter, and H. Ney, "A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition," in *ICASSP*, 2017.

[32] M. Zeineldeen, A. Zeyer, R. Schlüter, and H. Ney, "Layer-normalized LSTM for hybrid-HMM and end-to-end ASR," in *ICASSP*, 2020.

[33] S. Kanthak and H. Ney, "Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Orlando, FL, USA, May 2002.

[34] M. Killer, S. Stuker, and T. Schultz, "Grapheme based speech recognition," in *Eurospeech*, 2003.

[35] Y.-H. Sung, T. Hughes, F. Beaufays, and B. Strope, "Revisiting graphemes with increasing amounts of data," in *ICASSP*. IEEE, 2009.

[36] D. Le, X. Zhang, W. Zheng, C. Fügen, G. Zweig, and M. L.

[37] Seltzer, "From senones to chenones: Tied context-dependent graphemes for hybrid speech recognition," in *ASRU*, 2019.

[37] L. Breiman, J. Friedman, C. Stone, and R. Olshen, *Classification and regression trees*, CRC press, 1984.

[38] S. J. Young, "The general use of tying in phoneme-based HMM speech recognisers," in *ICASSP*, 1992.

[39] K. Irie, R. Prabhavalkar, A. Kannan, A. Bruguier, D. Rybach, and P. Nguyen, "On the choice of modeling unit for sequence-to-sequence speech recognition," in *Interspeech*, 2019.

[40] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[41] Y. Miao, M. Gowayyed, X. Na, T. Ko, F. Metze, and A. Waibel, "An empirical exploration of CTC acoustic models," in *ICASSP*, 2016.

[42] T. N. Sainath, R. Prabhavalkar, S. Kumar, S. Lee, A. Kannan, D. Rybach, V. Schogol, P. Nguyen, B. Li, Y. Wu, et al., "No need for a lexicon? evaluating the value of the pronunciation lexica in end-to-end models," in *ICASSP*, 2018.

[43] 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," in *ICONIP*, 2018.

[44] W. Wang, Y. Zhou, C. Xiong, and R. Socher, "An investigation of phone-based subword units for end-to-end speech recognition," Preprint arXiv:2004.04290, 2020.

[45] Y. Kubo and M. Bacchiani, "Joint phoneme-grapheme model for end-to-end speech recognition," in *ICASSP*, 2020.

[46] K. Hu, A. Bruguier, T. N. Sainath, R. Prabhavalkar, and G. Pundak, "Phoneme-based contextualization for cross-lingual speech recognition in end-to-end models," Preprint arXiv:1906.09292, 2019.

[47] E. Variani, D. Rybach, C. Allauzen, and M. Riley, "Hybrid autoregressive transducer (HAT)," in *ICASSP*, 2020.

[48] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, 2017.

[49] R. Collobert, C. Puhrsch, and G. Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," Preprint arXiv:1609.03193, 2016.

[50] W. Zhou, S. Berger, R. Schlüter, and H. Ney, "Phoneme based neural transducer for large vocabulary speech recognition," in *submitted to ICASSP*, 2021.

[51] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015.

[52] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*, 2016.

[53] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, 1997.

[54] A. Zeyer, T. Alkhouli, and H. Ney, "RETURNN as a generic flexible neural toolkit with application to translation and speech recognition," in *ACL*, 2018.

[55] D. Rybach, S. Hahn, P. Lehnen, D. Nolden, M. Sundermeyer, Z. Tüske, S. Wiesler, R. Schlüter, and H. Ney, "RASR - the RWTH Aachen University open source speech recognition toolkit," in *ASRU*, 2011.

[56] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *ICASSP*, 1992.

[57] T. Nguyen, S. Stueker, J. Niehues, and A. Waibel, "Improving sequence-to-sequence speech recognition training with on-the-fly data augmentation," Preprint arXiv:1910.13296, 2019.

[58] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. Soplin, R. Yamamoto, X. Wang, et al., "A comparative study on Transformer vs RNN in speech applications," in *ASRU*, 2019.

[59] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP*, 2015.