Principled Comparisons for End-to-End Speech Recognition: Attention vs Hybrid at the 1000-Hour Scale

Aku Rouhe, Tamás Grósz, and Mikko Kurimo, *Senior Member, IEEE*

***Abstract*—End-to-End speech recognition has become the center of attention for speech recognition research, but Hybrid Hidden Markov Model Deep Neural Network (HMM/DNN) -systems remain a competitive approach in terms of performance. End-to-End models may be better at very large data scales, and HMM/DNN-systems may have an advantage in low-resource scenarios, but the thousand-hour scale is particularly interesting for comparisons. At that scale experiments have not been able to conclusively demonstrate which approach is best, or if the heterogeneous approaches yield similar results.In this work, we work towards answering that question for Attention-based Encoder-Decoder models compared with HMM/DNN-systems. We present two simple experimental design principles, and how to build systems adhering to those principles. We demonstrate how those principles remove confounding variables related to both data, and neural architecture and training. We apply the principles in a set of experiments on three diverse thousand-hour-scale tasks. In our experiments, the HMM/DNN-systems yield equal or better results in almost all cases.**

***Index Terms*—ASR, End-to-End, HMM/DNN.**

I. Introduction

BROADLY, during the 1990 and 2000 decades, Hidden Markov Model (HMM)-based speech recognition systems were the only mainstream approach for building speech recognition systems. HMM-based speech recognition has developed in many ways, for instance, by using Deep Neural Network (DNN) acoustic models in Hybrid HMM/DNN-systems. The HMM/DNN-system approach continues to be competitive today, but during the 2010 decade, multiple other competitive approaches emerged, most common of them being: Connectionist Temporal Classification (CTC) [1], RNN-Transducers [2] (also called simply Transducers), and Attention-based Encoder-Decoder (AED) models [3], [4]. These competing approaches are all referred to by the umbrella term End-to-End speech recognition.

The End-to-End approaches have many differences with HMM/DNN-systems, but there are also many differences between the End-to-End approaches. Comparing these *heterogeneous* approaches is not straight-forward, since experimental results cannot be attributed to a single cause, because of the many differences. Broadly speaking, in very large data tasks, at the ten-thousand or hundred-thousand-hour scales, End-to-End models are reported to outperform HMM/DNN-systems [5], [6], and with limited resources, with a hundred hours and less, HMM/DNN-systems may have an advantage [7], [8]. This empirical result is also plausible theoretically, as End-to-End models generally rely less on in-built structure [9].

However, the thousand-hour scale is particularly interesting for comparisons, because no approach has been shown to be conclusively better [7], [10], and because the thousand-hour scale is still possible to reach, resource-wise, in multiple languages, in open datasets, and in multiple styles [11], [12], [13].

In this work we propose two simple experimental design principles, which allow making stronger statements from comparisons of heterogeneous speech recognition systems. The *Equal Data Setting* is a principle which avoids confounding differences in data. The *Matched Encoder Setting* is a principle which avoids differences in neural architecture and training.

We apply these principles in a set of experiments comparing HMM/DNN-systems and AED-models. We focus on these two approaches, because both have recently been shown to have high performance [10], and because they are very different: using input- vs. output-synchronous decoding, having implicit internal vs. explicit external language models, using hard and explicit vs. soft and implicit alignments. Additionally, building multiple well-performing speech recognition systems is a large effort, and thus we need to limit the scope of the work.

Our main contributions are as follows.

Firstly, we propose a conceptual framework for comparing End-to-End and HMM -based speech recognition systems. We develop the Equal Data Setting and Matched Encoder Setting principles for experimental design. We show how to build HMM/DNN-systems and AED-models, which adhere to these principles.

Secondly, we conduct a set of experiments comparing HMM/DNN-systems and AED-models, with the HMM/DNN-systems consistently reaching equal or better performance compared to our AED-models. Because of the principles we followed, we are able to say that the results are not due to having

Manuscript received 15 March 2023; revised 14 September 2023 and 9 November 2023; accepted 9 November 2023. Date of publication 24 November 2023; date of current version 8 December 2023. This work was supported in part by the Academy of Finland under Grants 345790 and 337073, in part by Business Finland Project LAREINA, in part by KAVI, and in part by Business Finland LAREINA under Grant 7817/31/2022. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Rohit Prabhavalkar. *(Corresponding author: Aku Rouhe.)*

The authors are with the Department of Information and Communications Engineering, Aalto University, 00076 Aalto, Finland (e-mail: aku.rouhe@aalto.fi; tamas.grosz@aalto.fi; mikko.kurimo@aalto.fi).

Digital Object Identifier 10.1109/TASLP.2023.3336517

additional data, nor due to the neural architecture or training favouring the HMM/DNN-systems.

Thirdly, we make multiple discoveries about our HMM/DNN-systems. We develop a multi-head decoding method, which yields the best results. We find that frame-level training is still useful, but on the other hand, expert pronunciation-lexicons and tree-clustering for state-tying do not appear necessary, echoing other recent work [14], [15], and potentially simplifying the HMM/DNN-system.

A. *Related Work*

Speech recognition approaches are most commonly compared by reporting various state-of-the-art results from the literature so far. On popular benchmarks this may also lead to intense competition, with rapid progress on the state-of-the-art numbers. For example, in [16], an AED-model using SpecAugment was reported to surpass state-of-the-art results on Librispeech, but concurrently in [7], an HMM/DNN-system achieved the lowest error rates at that point. In [17] an AED-model was found to outperform other Switchboard-300 results in the literature. Though these results were later surpassed by an HMM/DNN-system in [10], the results of the earlier work with an AED-model were already improved to the lowest currently known numbers in [18]. In this competition for the state-of-the-art numbers, the systems are not constrained in anyway, and implementations differ considerably, e.g. in terms of the number of training epochs.

Our proposed principled experimental design is more similar with experiments where speech recognition approaches are compared directly, applying some specific constraints. For example [19] compares HMM/DNN-systems, CTC-models, AED-models, and Transducer-models using the same encoder architecture for all, though exact training parameters are not described. In [7], AED-models and HMM/DNN-systems are compared using the same neural model type (Bidirectional LSTM), though the authors do not use exactly matching architectures nor training hyperparameters, but opt instead to optimize each model's recipe in isolation.

Concurrently with our work, [9] provides an overview survey of End-to-End speech recognition. The survey breaks down the term End-to-End into more precisely defined concepts, and also includes a section relating End-to-End speech recognition to HMM-based speech recognition (the term Classical speech recognition is used). Additionally, [20] presents the field from an industry perspective, stating how choosing an appropriate speech recognition approach to develop and deploy is not easy.

II. Principles for Comparisons of Heterogeneous Systems

Here we introduce two constraining principles for building heterogeneous speech recognition systems for direct comparisons. The goal is to create comparisons which reveal more about the differences in the speech recognition approaches, as opposed to confounding elements such as differences in data or optimization.

Firstly, we introduce the Equal Data Setting, which we first explored in [8]. Secondly, in this work we also propose the Matched Encoder Setting.

A. *Equal Data Setting*

Different speech recognition approaches may be able to leverage different data sources: hybrid HMM/DNN-systems are able to leverage curated pronunciation lexicons and additional text-only data, while standard End-to-End AED-models only use transcribed speech. In a practical, commercial application, where the end goal is to train the best performing model, it is sensible to use all available data sources. However, we wish to quantify the differences in the models - not the differences in data. Thus, we argue that the models should be compared under an Equal Data Setting [8], where the data that is available to each approach is exactly the same.

If we are comparing a speech recognition approach that only uses End-to-End Data, i.e. just transcribed speech, the Equal Data Setting limits all approaches to End-to-End Data. Mostly, this limitation affects HMM/DNN-systems. In an Equal Data Setting, HMM/DNN-systems use grapheme-based lexicons and transcript-based language models. Training language models only on transcripts will likely lead to less capable models, but requires no special techniques, as it is just a reduction in the amount of data available. Grapheme-based lexicons, where the acoustic model units are based on characters, can be used without any pronunciation dictionary data. For languages such as English, with non-trivial pronunciation, grapheme-based systems can be expected to perform slightly worse, whereas for languages like Finnish, which have a transparent orthography, grapheme-based systems are the norm and curated lexicons do not offer a benefit [21].

Instead of limiting the HMM/DNN-system to End-to-End Data, it is possible to extend End-to-End models to use other data types. Developing methods to leverage text-only data is probably beneficial in any speech recognition approach. Joint Training can be maintained by synthesizing audio or encoder representations for the text-only data [22], [23], however this requires an additional synthesis model. A simpler method is using an external neural Language Model (LM) in shallow fusion with an AED-model. This way it is possible to use additional text-only data and retain an Equal Data Setting. However, the resulting model is no longer Jointly Trained. Though shallow fusion is the standard approach, it does not compensate for the internal language model in the AED-model (only learnt on the transcripts of the data). This compensation is possible through more sophisticated methods (e.g. [24]). To leverage pronunciation dictionaries, AED-models can be made to use phoneme-based units, though this can make decoding more complicated and may not offer any benefit over grapheme-based units [25].

Speech recognition research has a long tradition of controlled benchmarks with clearly defined training, validation and test data (e.g. [11], [26], [27]). With the advent of End-to-End approaches, the Equal Data Setting is needed as a more precise definition for experiments, due to the benchmarks' additional text resources and pronunciation lexicons.

B. Matched Encoder Setting

Virtually all current ASR approaches use a notion of an Speech-Encoder, which maps the audio into a representation that contains only the information relevant for transcription. In an AED-model, the Speech-Encoder is simply the Encoder part of the model, and in an HMM/DNN-system, the Speech-Encoder is at the core of the Acoustic Model, before the representations are mapped into emission probabilities. Both approaches can use the exact same neural architectures for the Speech-Encoder. This similarity is contrasted by very different paths for decoding text. The AED-model includes the attentional decoder, from which text is produced output-synchronously. An HMM/DNN-system uses a search system that integrates the probabilities of the Acoustic Model, the hidden Markov model transition probabilities, a pronunciation model, a separate Language Model, and possibly other models.

Since the task of speech recognition remains the same regardless of approach, similar representations are probably useful in all approaches. Unfortunately the representations learned by the AED-model encoder have only been studied inextensively [28], [29].

We propose to use the same neural architecture for the Speech-Encoder in heterogeneous speech recognition systems, such as with HMM/DNN-systems and AED-models. This ensures that the Speech-Encoder has the same capacity and is equally good at modeling acoustics. In addition to the neural architecture, the important neural network training hyperparameters, such as batch size, learning rate schedule, and number of training epochs, are matched. Only the approach specific hyperparameters (e.g. the weight of an auxiliary Cross-Entropy loss in an HMM/DNN-system) have no counterpart and thus cannot be matched. The initialization is also matched. In similar vein, all speech recognition approaches probably benefit from using augmentation or auxiliary inputs such as speaker embeddings [16], [18], [30], [31], and their use should be matched.

We call this approach of using the same Speech-Encoder, with the same augmentation and auxiliary inputs, and the same training hyperparameters the Matched Encoder Setting.

In a general sense, the goal is to avoid the phenomenon where two systems are compared, but one of them is more heavily optimized – what could be called the *favourite child problem*. However, matched hyperparameter training is not trivial, since neural network training depends on the criterion: the same training hyperparameters could be closer to optimal for one approach. We still propose picking one model to optimize first, and then applying those parameters to the other system. Crucially, this sets a lower bound on the performance of the latter system – it could only improve through further optimization. Furthermore, if the latter system outperforms the former system, it is not a result of the favourite child problem (at least in terms of hyperparameter tuning). Finally, as we will show in Section II-B, hyperparameters which work well for one system are often applicable to another, as well.

C. What Else Should Be Controlled For?

There are certain things that we strived to control for, that do not clearly follow from the Equal Data Setting and Matched Encoder Setting. Firstly, we used the same subword vocabularies for the language models of the HMM/DNN-systems and for the outputs of the AED-models. Secondly, if we apply sequence-discriminative training, we should apply it to both approaches. Thirdly, we generally use single-pass decoding with both models, in this case restricting the HMM/DNN-system to N-gram language models, but we also explore neural language model rescoring in some experiments.

We believe the principles proposed in this section follow from good scientific practice. We also believe that following these principles allows us to draw stronger conclusions from our experiments in comparing HMM/DNN-systems and AED-models. Nevertheless, these principles cannot cover all design choices in determining the compared systems. We discuss limitations in Section V-A.

III. Speech Recognition Systems

We use the principles introduced in Section II to build comparable HMM/DNN-systems and AED-models. We aim to build HMM/DNN-systems and AED-models following well-established, modern practices.

The Matched Encoder Setting practically necessitates using the same software tools for training both the HMM/DNN-system acoustic model and the AED-model, since different tools can very easily have subtle differences in neural implementations. There are few public, open source tools that allow this readily. One example is combining the Returnn and the RASR toolkits [32] in the TensorFlow ecosystem. In the PyTorch ecosystem, the Espresso [33] and the k2-fsa[1] toolkits allow some form of AED-models and HMM/DNN-system, but both lack for example sequence-discrinative AED-model training and Gaussian Mixture Models (GMM). We conduct our experiments in the PyTorch ecosystem, and opt to use SpeechBrain [34] to train neural networks, and build the full recipes by integrating many different toolkits. We release our recipes online,[2] hoping to help further research in this implementation-intensive area.

We use three different Speech-Encoder neural architectures: the Convolutional-Recurrent-Feedforward (CRDNN) model [34], the Conformer (Confo) [35], and wav2vec 2.0 (w2v2.0) [36]. The CRDNN and Conformer-models take (respectively) 40- and 80-dimensional Mel-scale filter bank log-energy vectors as input. Both architectures have a front-end of two convolutional layers, with the CRDNN using layers of 64 and 128 channels, and the Conformer using layers of 64 and 32 channels, and with both architectures using 3-by-3 kernels. The convolutional layers subsample the input in time, three-fold for HMM/DNN-systems and four-fold for AED-models, resulting in 30 ms and 40 ms output frame-rates, respectively. This minor difference in the encoder does not change the number

[1] [Online]. Available: https://github.com/k2-fsa/icefall

[2] [Online]. Available: https://github.com/aalto-speech/equal-data-matched-encoder-experiments

of parameters, but the different ASR approaches simply work best at different time-granularities. A small exception is the projection layer after the Conformer convolutional front-end, which is necessarily slightly wider when the total stride is 3 (compared to 4 in the AED). From here on the architectures diverge. On the CRDNN, the convolutional layers are followed by three 512-wide bidirectional LSTM layers, and finally by one 512-wide feed-forward layer. The CRDNN encoder has 27 million parameters altogether. With the Conformer, we use two sizes: a small one (Conformer$_S$) and a large one (Conformer$_L$). Conformer$_S$ has 12 Conformer layers with width 144, 4 attention heads, and a feed-forward dimensionality of 1024. This results in an encoder of just 9.3 million parameters. Conformer$_L$ has 12 Conformer layers with width 512, 8 attention heads, and a feed-forward dimensionality of 2048. The large Conformer encoder has 76.2 million parameters.

The wav2vec 2.0 encoder is the *Large* size, which has 318 million parameters. The encoders have been pretrained on a large untranscribed speech datasets using the wav2vec 2.0 Self-Supervised Learning (SSL) approach. The model has a convolutional frontend, which takes the raw audio waveform as input. We keep the convolutional frontend parameters frozen. The bulk of the model is made up of Transformer (Trafo) layers. We use openly available pretrained parameters (Uralic V2 for Finnish, LV60 for Librispeech). The pretraining SSL approach is explained in [36]. On top of the wav2vec 2.0 pretrained model, we add two randomly initialized feed-forward layers (1024-wide), which slightly improved our results in preliminary experiments. The wav2vec 2.0 encoder natively runs at a 20 ms output frame-rate. Thus its output does not cleanly divide into the 30 ms rate of the HMM/DNN-system. Instead, the wav2vec 2.0 -based HMM/DNN-system uses the 20 ms output frame-rate. The AED-model simply takes every other output, yielding the regular 40 ms frame-rate.

The Conformer encoders are trained with SpecAugment [16], as it is part of the recipe we adopt. We make a small adjustment: we do not use on-the-fly time stretching, so that our original GMM-alignments can be used. Although purely sequence-trained models could use time-stretching, in preliminary tests, we found that removing the time stretching yields us the same results as the recipe we adopt. The original work on SpecAugment also suggests the time-stretching is not crucial [16]. The CRDNN and wav2vec 2.0 encoders do not use SpecAugment, keeping in line with our earlier work. Augmentation and SpecAugment yield better performance, but this applies to both AED-models [16] and HMM/DNN-systems [30], and as such we believe it to be mostly a matter orthogonal to comparisons such as those presented here.

A. Hybrid Hidden Markov Model / Deep Neural Network Systems

The main HMM/DNN-systems are built in many stages, starting with GMM acoustic models. These are then followed by DNN acoustic models which use both the Lattice-Free Maximum Mutual Information (LF-MMI) [37] and Cross-Entropy (CE) training criteria. Language models are trained separately. To study the different benefits that the GMM alignments yield, we also train HMM/DNN-systems which use either Flat Start (FS) DNN acoustic models or only use Cross-Entropy targets.

Our acoustic models use word-position-dependent grapheme-units (permitting four variations of a character: at the beginning, inside, and at the end of a word, and as single character words [38]), except in Section IV-B, where we additionally present results with word-position-dependent phoneme-units for contrast.

1) Gaussian Mixture Models: The hybrid HMM/DNN-system acoustic model recipe begins by training increasingly more complex GMMs. We use the Kaldi toolkit [39] for all GMMs. We follow the Kaldi standard four-stage GMM recipe outline, where the last stage is a speaker-adapted tri-unit tristate HMM/GMM-system. The final GMM acoustic model is used to align the training data. These alignments are then used both in the HMM tree-clustering algorithm for state-tying and for Cross-Entropy target labels.

2) Deep Neural Network Acoustic Models: The early DNN acoustic model formulations computed the probability of the input belonging to a particular HMM emission state, and turned this into emission likelihoods through normalizing (dividing) by the prior probability. These DNNs use the frame-wise Cross-Entropy criterion, which trains the network to match the GMM alignments.

Cross-Entropy training remains a mainstream DNN acoustic model training method, either used as the first training phase, or as an auxiliary task. However, to achieve state-of-the-art performance, HMM/DNN-systems use some form of sequence-discriminative training. We use the sequence-discriminative LF-MMI criterion, taking the implementation from PyChain [40]. Most of the improvement can be acheived with any sequence-discriminative criterion, but a criterion that directly minimizes the expected error, which we lack, could still yield some further improvements [37], [41]. This presents a small caveat in the interpretation of our results.

As recommended, we add l_2-regularisation with weight 0.0005 to the outputs of the LF-MMI head [37], yielding minor improvements. The outputs of the LF-MMI DNN acoustic models are typically interpreted as logarithmic pseudo-likelihoods, requiring no division by the prior. It has been shown that LF-MMI can be used for Flat Start training, requiring no alignments, an output space based on simple pruning instead of tree-based clustering, and starting from a randomly initialized neural network [42]. This allows pure sequence-level training, offering similar simplicity as CTC training of acoustic models, but with a sequence-discriminative criterion.

Our main HMM/DNN-systems experiments use Cross-Entropy and LF-MMI in a multi-task learning setup. For the Cross-Entropy loss we apply uniform Label Smoothing (LS), which can help calibrate the output of the model, aiding in beam search [43]. In multi-task learning, the Cross-Entropy and LF-MMI criteria have their own output head, which is a separate linear layer, though both heads use the same units. We note that since both heads are used to compute HMM emissions

TABLE I
HMM / DNN Development Results on Finnish Parliament (FP) Train20 and Librispeech, Using Transcript LMs, Ordered by Decreasing WER

Training Criteria	**Inference Outputs**	**Dev WER [%]**	
FP Train20		Dev16	
LF-MMI	LF-MMI	15.12	
LF-MMI + CE	LF-MMI	14.32	
FS LF-MMI	LF-MMI	14.29	
CE	CE	13.73	
LF-MMI + CE	CE	13.11	
LF-MMI + CE	LF-MMI + CE	**12.15**	
Librispeech		Clean	Other
CE	CE	5.34	14.44
LF-MMI + CE	CE	5.17	13.88
FS LF-MMI	LF-MMI	5.09	13.16
LF-MMI + CE	LF-MMI	4.82	12.66
LF-MMI + CE	LF-MMI + CE	**4.61**	**12.43**

FS stands for the flat start approach, which in this case means a trivially pruned tree (instead of state-tying based on GMM alignment information). All results with CRDNN encoders and (by exception) 80-element feature vectors.

likelihoods, perhaps the term multi-loss learning or multi-head model could also be appropriate. We use a three-fold reduced output frame-rate for both the Cross-Entropy and the LF-MMI heads, as is typical with LF-MMI. When we use the Cross-Entropy output head in inference, we normalize the output with a prior vector. The prior is estimated empirically by averaging the Cross-Entropy head outputs on a sample of the training data. We decode by computing log-likelihoods in SpeechBrain and then using beam search in the Kaldi Weighted Finite State Transducer (WFST) decoder.

The common Kaldi inference time solution is to discard the Cross-Entropy head, and only use the LF-MMI outputs. Instead, we find that in our implementation, the best performance is achieved by using both output heads and linearly combining their outputs after a log-softmax, with the same weights as used during training (0.1 weight for Cross-Entropy). To the best of our knowledge, this proposed multi-head inference is a novel improvement for the HMM/DNN approach, though it resembles an efficient form of model combination. We presented initial results using this approach in [44] and explore it here in more detail. Table I compares the various acoustic model training criteria and output heads used during inference. The LF-MMI + Cross-Entropy training has a clear benefit both over LF-MMI alone or Cross-Entropy alone on both Librispeech and Finnish Parliament Train20 (see Section IV-A for dataset information). On Finnish Parliament Train20, perhaps because of the extensively tuned HMM/GMM recipe, the Cross-Entropy head yields the best single-head results. On Librispeech, the LF-MMI head is the better one of the single output heads. Additionally, we find that the simple pruning Flat Start LF-MMI outperforms the tree-clustering state-tied LF-MMI in our test on Finnish Parliament Train20. This is surprising, but seems to suggest that the tree-clustering does not always yield better performance in HMM acoustic modeling, which is also suggested in other recent work [15] and would simplify the HMM/DNN-system further. The investigation of this phenomenon is out of scope for this work.

3) Language Models: HMM/DNN-systems typically use N-gram language models, as they can easily be made computationally feasible for single-pass decoding. Additionally, large neural language models may be used in rescoring to improve results. Under the End-to-End Data limitation, the amount of data available for language modeling is lower than in typical systems. This emphasizes the data sparsity problem inherent with N-gram language models using large vocabularies. Thus, it is especially important to use subwords as the language modeling unit, which leads to a smaller vocabulary. We use Byte Pair Encoding (BPE) units with SentencePiece segmentation. With SentencePiece units, we take care to handle the word-position-dependent units correctly [8]. With subword units, it is especially important to use longer N-gram spans [45] and thus we use the VariKN toolkit, which can grow large span modified Kneser-Ney backoff language models [46]. We use 10-gram models for all transcript-based language models.

Neural language models are commonly thought to be more data hungry than N-gram models. Thus the benefits from neural language model rescoring may be diminished under the End-to-End Data limitation. Nevertheless, we present some experiments using neural language models, which are trained on the transcripts only. All of our neural language models are based on the Transformer architecture and use the same subword units as the corresponding N-gram models. With HMM/DNN-systems, we apply these neural language models in 100-best list rescoring (we also tried a 1 000-best list but it did not improve results). The neural language models are implemented in SpeechBrain.

The language model weight and word-insertion-penalty are important decoding hyperparameters, and are optimized on development sets.

We note that it has been shown that with efficient implementations, arbitrary history length neural language models can be applied to single-pass search in HMM/DNN-systems [47]. However, here the single-pass HMM/DNN-system implementation is limited to local context language models, leaving the advantages of arbitrary-length history modeling to the AED-model in this comparison.

B. Attention-Based Encoder-Decoder Models

The AED-models add an attentional decoder on top of the Speech-Encoder. The decoder uses attention to find relevant parts in the input, and then computes a distribution over the output text units. We use the same set of subword units that the HMM/DNN-system language models use. To optimize the networks, we employ the Cross-Entropy criterion with label smoothing and add an auxiliary CTC criterion, which has its own output head on top of the encoder. The CTC head uses the same subword units as the main attentional decoder. For the CRDNN models, the CTC criterion is only used for the first 15 nominal epochs, the idea being to aid learning in the beginning of training, since the attention mechanism is difficult to learn from random initialisation. For the Conformer and wav2vec 2.0 encoders, we use hybrid CTC/Attention modeling, where the CTC outputs are also used in decoding [48]. This joint CTC/Attention decoding is somewhat symmetrically matched by the two-output-head decoding in the HMM/DNN-system. We decode with beam search in SpeechBrain. To deal with the length bias of AED-models [49], we use an end-of-sentence probability threshold and an attention coverage penalty [50].

Because the HMM/DNN-system uses sequence-discriminative training, we want to apply a sequence-discriminative criterion to the AED-model as well. We implement the Minimum Word Error Rate (MWER) -criterion [51]. We use the recommended settings: sampler beam size 4, Cross-Entropy as regularisation with weight 0.01, and regularisation through subtracting the mean number of errors on a sample. We find it is important to use word-level MWER, not subword-level, though a subword implementation is faster because it requires no Sentencepiece conversion.

We present some experiments where neural language models of Section III-A3 are used in shallow fusion with the AED-models.

C. Minimum Word Error Rate Training for Joint CTC/Attention Models

The classic MWER algorithm does not account for Joint CTC/Attention, so to use MWER, we needed to develop some additional solution. It could theoretically be possible to develop and implement MWER training for Joint CTC/Attention, but practically we deemed it out of the scope of this work. Another approach could be to use MWER training on the attentional decoder, and keep updating the CTC head with the regular CTC criterion. However, MWER training, which uses beam search at every step, is particularly compute intensive. Thus we deemed it best to freeze the encoder, and only update the attentional decoder with MWER. This way, after MWER finetuning, the encoder representations have not drifted away from ones learned by CTC, and we can again apply Joint CTC/Attention decoding.

IV. Experiments

We showcase how the Equal Data Setting and Matched Encoder Setting principles affect our results compared to not using such principles. The effects of the End-to-End Data limits placed in our proposed Equal Data Setting can be directly estimated with comparable systems, which do not conform to the Equal Data Setting. The effects of the proposed Matched Encoder Setting are seen indirectly.

In our main experiments, we compare AED-models and HMM/DNN-systems on three different tasks under the Matched Encoder Setting and the Equal Data Setting (MES-EDS Comparison). Though we are interested in the relative results of the different models, we also present some external baseline results where applicable. Section IV-E analyses the results. First, we introduce the datasets.

TABLE II
A Data Overview: Hours of Speech, Number of Speakers, Average Utterance Length, and Number of Words in the Transcription, for Each Data Subset

Data	Hours	Speakers	Avg.Len.	Words
Finnish Parliament Train20	1 783	302	6.5s	11M
Finnish Parliament Test16	5.7	21	10.6	35k
Finnish Parliament Test20	4.7	28	6.6s	27k
Librispeech Train	961	>1 000	12.3s	9M
Librispeech Test Clean	5.4	87	7.4s	52k
Librispeech Test Other	5.3	90	6.5s	52k
Combined Finnish Train	4 224	18 187	3.8s	29M
Lahjoita Puhetta Test	6.8	99	1.5s	55k
YLE Test	5.6	128	6.7s	37k

A. Datasets

We use three different thousand-hour-scale datasets: the Finnish Parliament Train20 dataset, Librispeech, and a Combined Finnish Data task. The Combined Finnish dataset includes both the full Finnish Parliament ASR Corpus and the Lahjoita Puhetta dataset, and it is a new task which we introduce here.

Table II summarises the data.

1) Finnish Parliament Train20: The Finnish Parliament (FP) ASR Corpus [52] is the largest publicly available transcribed Finnish speech corpus. The full training set is 3 087 hours, and has 449 different speakers. The transcripts consist of 19 million words. The speech, taken from recordings of the Finnish national parliament plenary sessions, is semi-spontaneous and covers a wide breadth of topics. The training data has two distinct subsets: Train16 and Train20. We pick the Train20 subset. The Train20 subset has an extensively tuned recipe for GMMs [52], which will aid in exploring the benefits of alignments in HMM/DNN-systems. The Train20 dataset has 1 783 hours of data from 302 speakers and its transcripts have 11 million words.

The Finnish Parliament ASR Corpus also has a development set, Dev16, and two test sets, Test16 and Test20. The Dev16 and Test16 sets are from the year 2016, and Test20 is from 2020. Additionally, the corpus has text-only resources for building language models. We use the 30 million word text dataset, and abbreviate it Parl30 M. This text data also derives from the transcripts of the plenary sessions, so it has some overlap with the training data. For all the Finnish Parliament Train20 experiments we use a 1 750 BPE unit vocabulary, which was deemed to work well in prior experiments [8].

2) Librispeech: Librispeech [11] is a well known and highly competed English read speech task. We use the 960 h full set from more than a thousand different speakers, and the transcripts have 9 million words. Librispeech has two development and test sets: a clean one and a noisy, “other” one. Official 4-gram language models and official pronunciation dictionaries are distributed alongside the data. The 4-gram language model has been trained on an official 800 million word text corpus. For the main experiments, we train our own language models on the speech transcripts, and use grapheme-based acoustic model units. We use a 5 000 units BPE unit vocabulary for all Librispeech experiments (this performed slightly better than 2 000 units in preliminary experiments).

We choose to experiment on Librispeech for a few reasons. Firstly, it is English, where expert-knowledge pronunciation dictionaries are typically used. Under the Equal Data Setting, we limit the HMM/DNN-systems to grapheme-units, and English allows us to quantify this side of the Equal Data Setting. Secondly, Librispeech has extensive baseline results for us to compare to. Lastly, Librispeech covers another style: read speech.

3) Combined Finnish Data: We combine two datasets, the Finnish Parliament ASR Corpus, and the Lahjoita Puhetta corpus [13], to form a new Combined Finnish task. This task uses the largest amount of transcribed Finnish speech training data published so far, to the best of our knowledge. This combined task is not just large, but also requires the speech recognition approaches to handle multiple styles and domains. Altogether the training set has 4 224 hours from 18 187 speakers, and the transcripts contain 29 million words.

The Finnish Parliament ASR Corpus is described above in Section IV-A1. The Lahjoita Puhetta corpus has 1 601 hours of transcribed speech from 17 821 different speakers. The speech was donated by the Finnish public and transcribed by professional transcription services. The speech is spontaneous and colloquial, covering many dialects and topics. A development and test set split was introduced in the original publication and we use the same setup in this work. The corpus includes automatically created time-alignments for the recordings. Since the original recordings are relatively long for speech recognition purposes, we split all the recordings by pauses, which were marked by the professional transcribers. This splitting is also done for the development and test sets, because the long recordings would lead to pathological output issues for AED-models [13]. These output issues require further research outside the scope of this work.

To create the Combined Finnish Data task development set, we simply combine the Lahjoita Puhetta and Finnish Parliament Dev16 development sets.

4) YLE Test: The YLE Test data contains about six hours of Finnish broadcast news speech. The test data has a corresponding development set, but we do not separately optimize parameters on it in this work.

B. Equal Data Setting

We evaluate how the Equal Data Setting affects our results by comparing models which differ in resources. We contrast transcript-only language model results with extra-text language model results. We compare grapheme-unit models with phoneme-unit ones. With these results, we showcase how important it is to decouple differences in data from differences in the ASR approaches.

On Finnish data, the only End-to-End Data limitation is the amount of language model data. In Table III we present experiments with HMM/DNN-systems using different language model data and language model setups and additionally we include results with AED-models with and without shallow fusion language models. We do not apply internal language model compensation here, which presents a caveat, as it could improve the language model integration. The comparisons with neural language models keep the data equal, but the AED-models are no longer jointly trained. The transcript-based Transformer language model yields a 4% relative improvement over the transcript 10-gram model for the HMM/DNN-system and a 13% relative improvement over the non-shallow-fusion result for the AED-model. The Parl30 M 10-gram and Transformer rescoring combination brings a 28% relative improvement over the transcript 10-gram HMM/DNN-system and a 36% improvement over the AED-model over the non-shallow-fusion result. The use of external language models is crucial to obtain the best speech recognition systems, and it is also important to develop better strategies and methods for language model integration in AED-models [24].

TABLE III
EXPERIMENTS VALIDATING THE EQUAL DATA SETTING ON FINNISH PARLIAMENT TRAIN20

Family	1st-pass LM	Rescoring LM	WER [%] Dev16	WER [%] Test16
HMM	Transcript 10-gram	-	11.72	8.21
		Transcript Trafo	11.12	7.90
		Parl30M Trafo	9.59	6.56
	Parl30M 10-gram	-	9.63	6.69
		Parl30M Trafo	8.61	5.94
AED	-	-	14.26	10.39
	Transcript Trafo		12.74	9.03
	Parl30M Trafo		10.00	6.66

All models use matched encoder setting CRDNNS.

TABLE IV
EXPERIMENTS VALIDATING THE EQUAL DATA SETTING ON LIBRISPEECH

LM	Units	Dev WER [%] Clean	Dev WER [%] Other	Test WER [%] Clean	Test WER [%] Other
Transcript BPE 10-gram	Graph.	4.66	12.68	4.76	12.95
Official Word 4-Gram		3.68	10.40	3.92	10.52
	Phon.	3.59	10.20	3.95	10.38

All models are HMM/ DNN-systems with matched encoder setting CRDNNS and (by exception) 80-element features.

On English data, under the End-to-End Data limitation, both extra text data and expert-curated pronunciation dictionaries are excluded. In Table IV we present results with HMM/DNN-systems, which use phoneme- or grapheme-units and systems which use the official 4-gram or our transcript-only language models. Creating English pronunciations for subword units is difficult, because the units should be pronounced differently depending on their context. Additionally, the segmentation is based on text compression, and as such does not take phonemic information into account. Therefore, our grapheme- and phoneme-unit comparison is performed with the official 4-gram word-level language model.

Additional text-only data is a clear benefit in speech recognition. The main novelty in our chosen Equal Data Setting is to compare AED-models with HMM/DNN-systems using transcript-based language models. Both the Finnish and English results validate that this yields a meaningful comparison, where the data difference is eliminated. Finally, on the English data, we see that phoneme-units may offer little to no benefit over grapheme-units, echoing similar results in [14].

TABLE V
Revisiting Results From Earlier Work, Which Used the Equal Data Setting, but Not the Matched Encoder Setting

Family	Encoder	Data	YLE Test WER [%]
Kaldi HMM [8]	TDNN	FP+SC+SD	28.07
ESPNet AED [8]	Transformer		27.84
Kaldi HMM [52]	TDNN	FP Train20	26.15
AED [52]	CRDNN		28.99
HMM	CRDNN	FP Train20	22.04
AED	CRDNN		26.41

The earlier work used finnish parliament train16, speecon, and speechdat (FP+SC+SD). The AED-models did not use language models, and the HMM/DNN-systems all used different transcript language models that were trained in the same way as those used in this work.

TABLE VI
Development Results With Matched Encoder Setting

Family	Batch	Iter.	Notes	Dev WER [%]	
Finnish Parliament Train20				Dev16	
AED	200s	200k	No LS	24.10	
	40s	1M	-	16.80	
HMM	200s	200k	No LS	12.67†	
	40s	1M	-	12.45†	
Librispeech				Clean	Other
AED	40s	1M	-	6.26	16.76
+Cont. above	+80s	+375k	-	5.14	15.04
AED	80s	1M	Big Dec	4.84	14.41
HMM	40s	1M	-	4.78	12.79
+Cont. above	+80s	+375k	-	4.53†	12.29†
HMM	80s	1M	-	4.57†	12.24†

All results with CRDNN encoders. No LS stands for no label smoothing. Dagger (†) marks statistically insignificant difference.

C. Matched Encoder Setting

We emphasize the importance of the Matched Encoder Setting by presenting various results from our experiments, as well as some external baselines. We show how not following the Matched Encoder Setting could lead to drawing the wrong conclusions.

First, we revisit some results that we previously presented on the YLE test data. In [8], we presented results under the Equal Data Setting, where the HMM/DNN-system and the AED-model have similar performance. The number of parameters and the use of auxiliary inputs were matched in that comparison. However, the AED-model used ESPNet Transformer-architecture recipe, whereas the HMM/DNN-system relied on a Kaldi TDNN-recipe. This leaves open the question of just how much did the AED-model gain over the HMM/DNN-system from using a more advanced neural model. In [52], we again used the Equal Data Setting, this time with a Kaldi HMM/TDNN-recipe compared against a CRDNN AED-model, using the Finnish Parliament Train20 data, with the Kaldi system outperforming the AED-model slightly. These latter results are in Equal Data Setting with our new results, presented in Table V. The new results show how under a Matched Encoder Setting, the HMM/DNN-system win over the AED-model is actually emphasized in this case. Furthermore, our new AED-model has roughly equal performance compared to the Kaldi HMM/TDNN on the Equal Data Setting comparison on Finnish Parliament Train20, which is further evidence that results not under a Matched Encoder Setting can be difficult to interpret. However, we note that the primary result in [8] was that the external text-only data is the key to improved results, which was shown by a clear margin, and served to emphasize the importance of using the Equal Data Setting.

In Table VI we report two initial sets of results from the course of performing these experiments. On the Finnish Parliament Train20 data, we tried two different combinations of batch size and number of updates, both adding up to seeing 40 million seconds of data. The HMM/DNN-system result change was statistically insignificant (by bootstrap estimate [53]), but the AED-model performed much better ($\approx 30\%$ relative) with 1 million iterations of 40 s batches. On Librispeech, we iteratively improved the AED-model, which leapt from 6.26% word error rate to 4.84% (a 23% relative improvement). The best AED-model surpassed the initial HMM/DNN-system, but with the changes in training hyperparameters, the HMM/DNN-system improved 4% relative, and still had the lowest error rate.

In Table V we showed how experiments using unmatched encoders can be difficult to interpret, particularly when the differences in error rates are small. It was important to apply the best AED-model neural modeling to the HMM/DNN-system as well - otherwise it might have seemed as if the HMM/DNN-system was outperformed. In Table VI we showed how the HMM/DNN acoustic model learning is not highly dependent on training hyperparameters, and the best AED-model training parameters also yielded the best HMM/DNN-system.

D. Comparison Experiments

We compare our best AED-models and HMM/DNN-systems under the Matched Encoder Setting and Equal Data Setting in three tasks. We first optimize the AED-model, and then apply the same hyperparameters to the corresponding HMM/DNN-system. As explained in Section II-B, this sidesteps the favourite child problem: the neural model optimization at least does not favour the HMM/DNN-system.

Each task tests ASR models in slightly different conditions and in the interest of making the set of experiments manageable to run, we do not test every model and approach on every task. On the Finnish Parliament Train20 and Librispeech tasks we optimize CRDNN and wav2vec 2.0 recipes. Additionally, on Librispeech, we adapt and apply a recently published well-performing Conformer recipe from SpeechBrain. On the Combined Finnish Data, we use the best CRDNN models from Librispeech.

1) Finnish Parliament Train20 Experiments: On Finnish Parliament Train20, we started with the AED-model from [52]. The CRDNN encoder is described in Section III. The attentional decoder was a single 512-wide Gated Recurrent Unit (GRU) layer, and used location-and-content aware attention. The system was trained for 100 nominal epochs, where each epoch had 10 000 updates on dynamically sized batches, targeting 40 seconds of audio per batch. The system is trained

TABLE VII
MES-EDS Comparison Experiments on the Finnish Parliament Train20 Data

				WER [%]		
Family	**Encoder**	**1st-pass LM**	**Rescoring LM**	Dev16	Test16	Test20
Kaldi HMM [52]	TDNN	Transcript BPE 10-gram	-	14.19	10.52	8.84
HMM	CRDNN	Transcript BPE 10-gram	-	11.72	8.21	7.59
FS HMM				14.18	10.55	9.39
AED		-		14.26	10.39	8.57
HMM	CRDNN	Parl30M BPE 10-gram	Parl30M BPE Transformer	8.61	5.94	8.23
FS HMM				9.26	6.52	9.63
AED		Parl30M BPE Transformer	-	10.00	6.66	7.40
HMM	wav2vec 2.0	Transcript BPE 10-gram	-	9.06	6.46	6.48
AED/CTC		-		9.72	6.79	6.61
HMM	wav2vec 2.0	Parl30M BPE 10-gram	Parl30M BPE Transformer	8.15	5.92	8.02
AED/CTC		Parl30M BPE Transformer	-	8.48	6.02	7.09

FS stands for the flat start approach.

with Adam using a 0.0001 learning rate, without learning rate scheduling.

This initial system was slightly improved by doubling the attention context vector size. We also tried using multi-headed attention, using a larger decoder, not using label smoothing, and trading number of steps for a larger batch size (same amount of data seen overall), but these did not improve results in our implementation. Further improvements were found by training more (75 additional nominal epochs of 5 000 updates) with larger batches (80 s), using a NewBob learning rate schedule. Finally, we improved the AED-model through sequence-discriminative finetuning with MWER training. The MWER finetuning only needed a few thousand steps to reach the best performance. At that point, the resulting AED-model had reached parity with the Kaldi HMM/TDNN baseline from [52]. However, a Matched Encoder Setting HMM/DNN-system outperformed the AED-model.

The wav2vec 2.0 models require less training to reach good performance. Since they are considerably more compute-expensive, we train them for 25 nominal epochs (10 000 updates, 40 s batches). We use NewBob learning rate scheduling throughout training. After this, the AED-model is also slightly improved through our modified MWER finetuning approach as described in Section III-C.

We note that MWER training is a sequence-discriminative finetuning step, while the HMM/DNN-system uses the sequence-discriminative LF-MMI criterion throughout training. The MWER finetuning only took a few thousand steps, so on Finnish Parliament, we did not match this training step with anything on the HMM/DNN-system side. We could have continued the regular HMM/DNN-system training for a few thousand more steps to match the training length exactly, but the HMM/DNN-system had already converged, so it would not have changed the results. However, we note that some criterion minimizing the expected error could have been used here.

On the Finnish Parliament data, we also experimented with additional text resources. We present results using a 12-layer Transformer neural language model trained for 200 nominal epochs (with early stopping) on the transcripts, and another one trained on the Parl30 M text. We also train a single-pass-capable 10-gram language model on the Parl30 M data.

TABLE VIII
The Finnish Parliament Train20 Language Model Perplexities Normalized to the Word Level

	Perplexity		
Language model	Dev16	Test16	Test20
Transcript BPE 10-gram	1307.78	1604.77	1424.81
Transcript BPE Transformer	1122.20	1313.82	1154.50
Parl30M BPE 10-gram	387.93	489.67	1463.00
Parl30M BPE Transformer	224.46	192.93	1582.59

All models permit an open vocabulary.

TABLE IX
Detailed Look At the Finnish Parliament Train20 Model Sizes

	Parameter count	
Submodel	CRDNN	w2v2.0
Encoder	20.8M	318M
LF-MMI Head	1.19M	2.54M
CE Head	1.19M	2.54M
FS LF-MMI Head	579k	-
Transcript BPE 10-gram	6.83M	
Parl30M BPE 10-gram	19.2M	
CTC Head	898k	1.79M
AED Head	898k	898k
Att. Decoder	8.85M	10.9M
Transcript BPE Transformer	88.3M	
Parl30M BPE Transformer	88.3M	

The Finnish Parliament Train20 results are reported in Table VII. Language model perplexities (normalized to the word level) are shown in Table VIII. We remind readers less familiar with Finnish, that the Finnish absolute perplexity values are often much higher than e.g. English, due to the much larger vocabulary. Model sizes are reported in Table IX. The N-gram language model parameter counts are measured by the number of N-grams in the model, though there may be both a probability and a backoff weight associated with it.

2) Librispeech Experiments: On Librispeech we start with the best AED-model configuration from Finnish Parliament Train20. We improve the initial Librispeech AED-model with a 1024-wide decoder and training with an 80-second batch size from the start for 100 nominal epochs of 10 000 updates. After the 60th epoch, we use a NewBob learning rate schedule. This equals seeing the full data a little over 23 times. We also try a larger batch size, an even larger decoder, different learning rates and learning rate schedules including warm-up, an LSTM decoder, and lowering the label smoothing value, but the changes listed above yield our best model. On Librispeech we use the same wav2vec 2.0 configuration as Finnish Parliament Train20, except we increase the batch size to 180 seconds, which helps slightly.

We adapt the SpeechBrain Conformer$_L$[3] and Conformer$_S$[4] optimized recipes to our data pipeline. Unlike our other recipes, the Conformers use SpecAugment and large batch sizes (2520 seconds). The models train for 120 nominal epochs of 1824 updates (so that nominal epochs approximately match full dataset epochs), using Noam learning rate scheduling with a warm-up, and the AdamW optimizer [54]. Unlike the CRDNN and wav2vec 2.0 AED-models, which use a GRU decoder, the Conformer recipes use a 6-layer Transformer decoder.

At decode-time the Conformer recipe uses 10 checkpoint parameter averaging [55], which improves results. However, the recipe does not include an MWER finetuning step, which we add. We decide to add MWER finetuning after parameter averaging, because our MWER finetuning is run for relatively few steps (20 nominal epochs of 200 updates, about 2 full epochs), which does not yield meaningfully different checkpoints to average. On all other results, MWER appears to provide a modest improvement, except with the Conformer$_L$ AED-model on Test Other. Furthermore, we verified that the improvements are from MWER and not just any training after parameter averaging: regular AED-model training after parameter averaging does not improve the results, as shown with the Conformer$_S$ AED-model results.

Since the MWER finetuning happens after parameter averaging (a non-standard approach), we match this on the HMM/DNN-system side with regular training for an equivalent amount of steps. The AED-model encoder is frozen, while the HMM/DNN-system encoder is not - this may be a small mismatch, but this way, the results more conclusively show that the HMM/DNN-system finetuning after parameter averaging is not an important step in our experiments. We only find very small improvements in the Conformer$_S$ results and the Flat Start Conformer$_L$ results, but not the main Conformer$_L$ HMM/DNN-system. We decide to report results both with and without MWER finetuning and continued training after parameter averaging, because the analysis in Section IV-E reveals that most of the improvements from training after parameter averaging do not stand credibility inspection.

For the CRDNN and wav2vec 2.0 models, we find that MWER finetuning does not yield any improvement. We believe MWER training has not found wide use on Librispeech even though it is a highly competed dataset, though [56] and [57] report minor improvements with it.

The Librispeech results are reported in Table X. We have included many relevant results published elsewhere. We believe that as small CRDNN models, our results are reasonable, beating the Kaldi results and falling behind Returnn systems that also use classic (non-Transformer) neural layers. Further improvements to our results might be found through a combination of larger models, longer training with more complex optimisation (such as curriculum learning), augmentation, and additional language models. Our Conformer$_L$ AED-model falls slightly behind a comparable ESPnet model, which may be partly explained by the ESPnet advanced S4 Decoder [58]. Our HMM/DNN-system wav2vec2.0 results roughly match the Clean results obtained in the original wav2vec 2.0 publication, but fall slightly behind on the Other data, most likely due to us not using any augmentation, and the original paper applying SpecAugment, which is very beneficial on Librispeech [16].

The Librispeech language model perplexities are listed in Table XI and a detailed look at the model sizes in Table XII.

3) Combined Finnish Data Experiments: Finally, we use the Librispeech CRDNN recipes on the Combined Finnish data. Since the Combined Finnish Data is computationally demanding, we decide to limit the Combined Finnish experiments to CRDNN models.

Our chosen Equal Data Setting places an upper bound on the language model data: the speech transcripts. However, because language model training is decoupled from acoustic model training in HMM/DNN-systems, we are able to further limit the text data to one domain only. We try limiting the language model data to either Finnish Parliament transcripts or Lahjoita Puhetta transcripts, to see if those models work better on their own domains. The Combined Finnish Data results are reported in Table XIII. The language model perplexities are shown in Table XIV and a detailed rundown of the parameter counts in Table XV.

E. Analysis of Results

We analyze the test results of the Finnish Parliament Train20 and Librispeech tasks in detail. In addition to WER, we briefly looked at Character Error Rate results, but they appeared to draw the same picture as the word-level results, and we decided to focus on WER in this work. Table XVI highlights key comparisons. A more comprehensive table of comparisons is published online.[5] We use a bootstrap estimate to measure how credible it is that the winning system is truly better [53], treating the 95% mark as a cutoff. To measure the extent to which the compared systems produce similar output, we compute three quantities. Sentence Difference is simply the percentage of utterances that resulted in different outputs from the two systems. Additionally, we compute Kendall's rank correlation coefficient, tau (τ) [60], a measure of ranking agreement, on both utterance and speaker-level WER. A tau value close to one indicates the same utterances or speakers were (in relative terms) easy or difficult for both

[3] [Online]. Available: https://github.com/speechbrain/speechbrain/blob/develop/recipes/LibriSpeech/ASR/transformer/hparams/conformer_large.yaml

[4] [Online]. Available: https://github.com/speechbrain/speechbrain/blob/develop/recipes/LibriSpeech/ASR/transformer/hparams/conformer_small.yaml

[5] [Online]. Available: https://github.com/aalto-speech/equal-data-matched-encoder-experiments/blob/main/full-pairwise-analysis.pdf

TABLE X
MES-EDS COMPARISON EXPERIMENTS ON LIBRISPEECH

	Encoder				**Dev WER [%]**		**Test WER [%]**	
Family	**Type**	**Size**	**LM**	**Notes**	Clean	Other	Clean	Other
Kaldi HMM[a]	TDNN	17M	Official Word 4-Gram	Perturbation, i-Vec., Phon.	3.87	10.22	4.17	10.62
Returnn HMM [7]	LSTM	180M[b]		Phon.	3.4	8.3	3.8	8.8
Returnn AED [7]	LSTM	Unknown	-		4.3	12.9	4.4	13.5
CTC [36]	wav2vec 2.0	318M		SSL, SpecAugment	2.1	4.5	2.2	4.5
ESPnet AED/CTC [58][c]	Conformer	76.2M		SpecAugment, S4 Decoder	2.07	5.31	2.29	5.13
Transducer [59][d]	Conformer	1B	Transformer	SSL, NST, SpecAugment	1.3	2.4	1.4	2.5
HMM	CRDNN	20.8M	Official Word 4-Gram	-	3.64	9.85	3.89	10.45
FS HMM					4.01	10.60	4.41	11.11
HMM	Conformer$_S$	9.30M	Official Word 4-Gram	SpecAugment, FT[e]	2.81	6.49	3.19	6.81
HMM	Conformer$_L$	76.2M	Official Word 4-Gram	SpecAugment	2.24	5.31	2.64	5.62
FS HMM				SpecAugment, FT	2.63	5.70	2.91	5.96
HMM	CRDNN	20.8M	Transcript BPE 10-gram	-	4.57	12.24	4.56	12.57
FS HMM					5.12	13.30	5.35	13.68
AED			-		4.84	14.41	5.11	15.04
HMM	Conformer$_S$	9.30M	Transcript BPE 10-gram	SpecAugment	3.50	8.21	3.57	8.28
AED/CTC			-	SpecAugment	3.46	8.94	3.63	8.89
HMM	Conformer$_S$	9.30M	Transcript BPE 10-gram	SpecAugment, FT	3.45	8.12	3.56	8.20
AED/CTC			-	SpecAugment, FT	3.46	8.91	3.64	8.92
AED/CTC				SpecAugment, MWER	3.42	8.83	3.60	8.85
HMM	Conformer$_L$	76.2M	Transcript BPE 10-gram	SpecAugment	2.49	6.07	2.67	6.11
FS HMM				SpecAugment	3.03	6.64	3.08	6.63
AED/CTC			-	SpecAugment	2.34	5.86	2.85	5.94
HMM	Conformer$_L$	76.2M	Transcript BPE 10-gram	SpecAugment, FT	2.52	6.15	2.69	6.14
FS HMM				SpecAugment, FT	2.96	6.64	3.00	6.63
AED/CTC			-	SpecAugment, MWER	2.32	5.87	2.64	6.03
HMM	wav2vec 2.0	318M	Transcript BPE 10-gram	SSL	2.16	5.15	2.24	5.05
AED/CTC			-	SSL	2.39	5.93	2.41	6.07

[a] From: https://github.com/kaldi-asr/kaldi/blob/master/egs/librispeech/s5/RESULTS, Phon. stands for phoneme units.
[b] Size mentioned in [14], Corresponding AED size not mentioned, but is similar.
[c] Encoder size estimated, as hyperparameters similar to our Conformer$_L$. S4 Decoder stands for Structured State Space Decoder.
[d] The best reported results we are aware of. NST stands for Noisy Student Training.
[e] FT stands for training after 10-checkpoint parameter averaging, which matches MWER finetuning.
FS stands for the flat start approach.

TABLE XI
LIBRISPEECH LANGUAGE MODEL PERPLEXITIES NORMALIZED TO THE WORD LEVEL, WITH RATE OF OUT-OF-VOCABULARY WORDS (OOVS) MARKED

	Dev		Test	
	Clean	Other	Clean	Other
Language model	**Perplexity**			
Transcript BPE 10-gram	303.83	285.67	308.54	285.01
Official Word 4-gram	151.65	140.61	158.13	145.75
	OOV Rate [%]			
Transcript BPE 10-gram	0.00	0.00	0.00	0.00
Official Word 4-gram	0.33	0.55	0.40	0.51

Note: since the language models use different vocabularies, the perplexity differences are only an approximate measure.

TABLE XII
DETAILED LOOK AT THE LIBRISPEECH MODEL SIZES

	Parameter count			
Submodel	CRDNN	Confo$_S$	Confo$_L$	w2v2.0
Encoder	20.8M	9.30M	76.2M	318M
LF-MMI Head	1.16M	327k	1.16M	2.30M
CE Head	1.16M	327k	1.16M	2.30M
FS LF-MMI Head	678k	-	678k	-
Transcript BPE 10-gram	3.61M			
Official Word 4-gram	145M			
CTC Head	2.57M	725k	2.57M	5.13M
AED Head	5.13M	725k	2.57M	5.13M
Att. Decoder	17.9M	2.58M	27.8M	19.9M

systems. A tau close to zero indicates the systems succeeded and failed on completely unrelated utterances or speakers. This type of agreement computation is naturally dependent on data, but comparing agreement values within a particular test set is possible.

We find that the CRDNN and wav2vec 2.0 HMM/DNN-systems consistently outperform corresponding AED-models, even under the Matched Encoder Setting and Equal Data Setting comparison. An exception is the Finnish Parliament Test20 data with Parl30 M language models. On Test20, the Parl30 M language models have higher perplexity (see Table VIII) and thus hurt performance (also seen in [52]), and this effect is larger on HMM/DNN-systems. Another exception is the wav2vec 2.0 system performance with the Parl30 M language models on Finnish Parliament Test16 data, which yielded roughly equal performance (credibility $\approx 79\%$).

The Conformer encoder comparisons all lead to approximately equal performance (credibility $< 95\%$). Besides the

TABLE XIII
MES-EDS COMPARISON EXPERIMENTS ON THE COMBINED FINNISH DATA

		WER [%]					
		Finnish Parliament			Lahjoita Puhetta		YLE
Family	**LM**	Dev16	Test16	Test20	Dev	Test	Test
HMM	All Transcript BPE 10-gram	11.43 (11.16)	7.97 (7.82)	7.78 (7.53)	22.00 (21.93)	25.28 (25.17)	24.04
HMM	FP Transcript BPE 10-gram	11.67 (11.12)	8.17 (7.93)	7.90 (7.70)	29.28 (28.95)	32.99 (33.06)	25.75
HMM	LP Transcript BPE 10-gram	16.46 (15.65)	11.89 (11.00)	10.87 (10.19)	22.35 (22.01)	25.90 (25.31)	29.88
AED	-	12.76	8.69	8.14	23.19	26.82	24.76

All rows use the CRDNN encoder. HMM WER in brackets is with LM weight optimized on current data development set, for comparison between LMs.

TABLE XIV
PERPLEXITIES OF THE COMBINED FINNISH 10-GRAM BPE LANGUAGE MODELS TRAINED ON TRANSCRIPTS OF EITHER THE COMBINED DATA (ALL), THE FINNISH PARLIAMENT (FP) SUBSET (DIFFERENT FROM TRAIN20), OR THE LAHJOITA PUHETTA (LP) SUBSET

	Perplexity					
	Finnish Parliament			Lahjoita Puhetta		YLE
Set	Dev16	Test16	Test20	Dev	Test	Test
All	1012.22	1361.00	1189.18	669.82	695.68	17.9k
FP	954.02	1257.96	1241.19	18.3k	18.9k	28.1k
LP	55.5k	74.5k	61.1k	562.04	562.04	165.7k

The perplexities normalized to the word level. All models permit an open vocabulary. Some perplexities are very large, note thousands marked with *k*.

TABLE XV
DETAILED LOOK AT THE COMBINED FINNISH MODEL SIZES

Submodel	**Parameter count**
CRDNN Encoder	20.8M
LF-MMI Head	1.47M
CE Head	1.47M
All Transcript BPE 10-gram	12.9M
FP Transcript BPE 10-gram	9.30M
LP Transcript BPE 10-gram	3.62M
CTC Head	2.57M
AED Head	5.13M
Att. Decoder	17.9M

encoder architecture, the difference to the CRDNN and wav2vec 2.0 experiments is that the Conformer recipes used very large batch sizes and needed to train for much longer. Another difference is that the Conformer recipes used SpecAugment, but we verified that this alone does not allow the AED-model to reach parity. Adding SpecAugment to the CRDNN AED-model recipe on Finnish Parliament improved the WER only from $14.36\% \rightarrow 13.64\%$ on Dev16, $10.39\% \rightarrow 9.99\%$ on Test16, and $8.57\% \rightarrow 8.47\%$ on Test20, which is still worse than the HMM/DNN-system without SpecAugment (having WERs of $11.72\%, 8.21\%, 7.59\%$ on those evaluation sets respectively). A final difference is that the Conformer AED-models used Transformer decoder layers, which may be beneficial in particular in conjunction with large batch sizes and longer training.

The wav2vec 2.0 systems consistently outperform both CRDNN and Conformer systems, with an exception on HMM/DNN-systems on Finnish Parliament Test16 when using Parl30 M language models, which has roughly equal performance (credibility $\approx 54\%$). The Parl30 M language models appear to provide the most improvement on that data. Another exception is the Librispeech Other data, where the wav2vec 2.0 AED-model has roughly equal performance with the Conformer$_L$ AED-model (credibility $\approx 59\%$).

The Flat Start HMM/DNN-systems on the other hand have more varied comparisons. With transcript language models, Flat Start CRDNN HMM/DNN-systems have roughly equal (credibility 79%) performance with AED-models on Finnish Parliament Test16. Librispeech Test Clean and Finnish Parliament Test20 lead to AED-model wins, and Librispeech Test Other to a Flat Start HMM/DNN-system win. With larger language models, the CRDNN Flat Start HMM/DNN-system has equal performance with the AED-model on Finnish Parliament Test16 (credibility 79%) and the Librispeech test sets and loses on Finnish Parliament Test20, due to the aforementioned Parl30 M phenomenon. The Flat Start Conformer$_L$ HMM/DNN-system loses to the AED-model on both Librispeech test sets. Finally, in this work Flat Start HMM/DNN-systems lose to regular HMM/DNN-systems without exception (though contrary results exist in the literature [61]).

We computed errors on rare words, in this case words that do not appear in the training data transcripts. We used Levenshtein alignments to find instances where rare words in the reference resulted in substitutions and deletions. Table XVII lists the results. The AED-model has slightly higher error rates on the rare words than the corresponding HMM/DNN-system in every comparison except with Conformer$_L$ encoders on Librispeech Test Other. Yet the Flat Start HMM/DNN-system performs slightly worse than the AED-model on all comparisons except with CRDNN encoders on Librispeech Test Clean. Perhaps the frame-level training in the main HMM/DNN-systems gives it an edge in modeling an unfamiliar acoustic sequence. However, the differences are not very large in any comparison.

We looked into word error streaks – how often do the systems have multiple consecutive edit operations. Fig. 1 plots the ratios of AED streaks to HMM streaks for streak lengths upto four. Longer streaks are too rare to yield meaningful data. With CRDNN encoders, the pattern is especially clear: AED appears to have relatively more longer streaks than the transcript HMM/DNN counterpart. This pattern is also visible with Conformers (particularly on Librispeech Test Clean), although for Conformers the result is less significant since there are no results for the Finnish data. Additionally, the Conformer figures show the effect of MWER finetuning, which appears to decrease long streaks of errors. The pattern of AED-models having more longer streaks than HMM/DNN-systems is not seen with wav2vec 2.0 encoders.

TABLE XVI
SELECTED PAIRWISE COMPARISONS ON THE TEST SETS, WITH (ITALIC) COMMENTS HIGHLIGHTING THE RESULTS INTERSPERSED

	A			*B*			WER Difference (*B*-*A*)			S-Diff.	WER Rank τ	
	Family	Encoder	LM Data	Family	Encoder	LM Data	[pp.]	Relative	Credib.	[%]	Utt.	Spk.
FP-16	*The FS HMM has approximately equal overall WER with the AED, though models produce different output for > 50% of utterances.*											
	FS HMM	CRDNN	Transcript	**AED**	CRDNN	-	-0.16	-1.6%	79%	72.67	0.47	0.80
	FS HMM	CRDNN	Parl30M	**AED**	CRDNN	Parl30M	+0.13	+2.0%	79%	53.48	0.54	0.83
	The Parl30M language model is key to the best performance on FP-16.											
	HMM	w2v2.0	Parl30M	**AED/CTC**	w2v2.0	Parl30M	+0.09	+1.7%	79%	42.84	0.65	0.87
	HMM	**w2v2.0**	Parl30M	HMM	**CRDNN**	Parl30M	+0.01	+0.2%	54%	40.04	0.66	0.88
FP-20	*On FP-20, the wav2vec 2.0 systems have equal performance.*											
	HMM	w2v2.0	Transcript	**AED/CTC**	w2v2.0	-	+0.12	+1.8%	82%	28.31	0.68	0.85
Librispeech Test Clean	*The FS HMM is close in WER with the AED, though models produce different output for > 50% of utterances.*											
	FS HMM	CRDNN	Transcript	**AED**	CRDNN	-	-0.24	-4.7%	97%	51.15	0.49	0.65
	The Conformer encoders lead to equal performance on Test Clean.											
	HMM	Confo$_S^{FT}$	Transcript	**AED/CTC**	Confo$_S^{MW}$	-	+0.04	+1.2%	69%	36.45	0.58	0.67
	HMM	Confo$_L$	Transcript	**AED/CTC**	Confo$_L^{MW}$	-	-0.03	-1.2%	67%	27.44	0.61	0.71
	HMM	Confo$_L$	**Orig. 800M**	**AED/CTC**	Confo$_L^{MW}$	-	-0.01	-0.2%	52%	30.95	0.53	0.61
	The post-parameter-averaging MWER finetuning or HMM/DNN training only yield credible benefits for Conformer$_L$ AED/CTC.											
	HMM	**Confo**$_S^{FT}$	Transcript	HMM	**Confo**$_S$	Transcript	+0.02	+0.4%	69%	6.87	0.94	0.92
	FS HMM	Confo$_L^{FT}$	Transcript	FS HMM	Confo$_L$	Transcript	+0.08	+2.7%	95%	17.44	0.80	0.86
	AED/CTC	**Confo**$_S^{MW}$	-	AED/CTC	**Confo**$_S$	-	+0.02	+0.6%	75%	9.16	0.92	0.90
	AED/CTC	**Confo**$_L^{MW}$	-	AED/CTC	**Confo**$_L$	-	+0.21	+7.4%	100%	13.47	0.84	0.81
Librispeech Test Other	*The HMM is slightly better than the AED with the smaller Conformer encoder, but the large one leads to equal performance on Test Other.*											
	HMM	Confo$_S$	Transcript	**AED/CTC**	Confo$_S$	-	+0.61	+6.9%	100%	59.65	0.57	0.75
	HMM	Confo$_L$	Transcript	**AED/CTC**	Confo$_L$	-	-0.17	-2.8%	94%	45.93	0.64	0.79
	The wav2vec 2.0 and the Conformer$_L$ encoder lead to equal AED/CTC performance.											
	AED/CTC	**w2v2.0**	-	AED/CTC	**Confo**$_L$	-	-0.13	-2.2%	59%	51.72	0.57	0.76
	The official 4-gram LM lets the Conformer$_L$ HMM overtake and the FS HMM catch up with the AED - but leads to unequal data.											
	HMM	Confo$_L$	**Orig. 800M**	**AED/CTC**	Confo$_L$	**-**	+0.32	+5.5%	100%	47.91	0.58	0.71
	FS HMM	Confo$_L^{FT}$	Transcript	AED/CTC	Confo$_L$	-	-0.69	-12%	100%	49.17	0.61	0.76
	FS HMM	Confo$_L^{FT}$	**Orig. 800M**	**AED/CTC**	Confo$_L$	**-**	-0.02	-0.3%	55%	51.75	0.52	0.67
	The post-parameter-averaging MWER finetuning or HMM/DNN training only yield credible benefits for Confo$_S$ HMM.											
	HMM	**Confo**$_S^{FT}$	Transcript	HMM	**Confo**$_S$	Transcript	+0.08	+1.0%	96%	16.40	0.91	0.93
	FS HMM	**Confo**$_L^{FT}$	Transcript	FS HMM	**Confo**$_L$	Transcript	-0.00	-0.1%	52%	34.60	0.75	0.85
	AED/CTC	**Confo**$_S^{MW}$	-	AED/CTC	**Confo**$_S$	-	+0.04	+0.5%	75%	21.61	0.89	0.93
	AED/CTC	**Confo**$_L^{MW}$	-	AED/CTC	**Confo**$_L$	-	-0.09	-1.5%	94%	19.60	0.87	0.91

Credibility (Credib.) is probability of improvement (of the better system) measured by bootstrap estimate [53]. The comparisons not shown here all have credibility> 95%. Sentence difference (S-diff.) is the percentage of transcriptions that were different between *A* and *B*. The superscripts ^{MW} and ^{FT} mean the post-10-model-averaging minimum word error rate finetuning and HMM finetuning, respectively. FS stands for the flat start approach. Finnish parliament test16 and test20 are shortened to FP-16 and FP-20.

More sophisticated statistical study of this phenomenon is left as future work.

We also looked into relative WER on the shorter and longer test set halves separately, but found it consistent across all speech recognition systems. On Finnish Parliament, the longer half had slightly higher WERs, whereas on Librispeech, the shorter half had higher WERs. We looked at largest wins (by number of edits) at the utterance level when comparing systems. This way we found some individual utterances, which lead to one system failing, with the other succeeding, proving the issue was not due to the utterance itself. In some cases the AED-model drops large portions of the utterance. We also find one case where the AED-model produces pathological repetitive output. Both are likely due to a failure of the attention mechanism.

In the Combined Finnish Data results in Table XIII the HMM/DNN-system outperforms the AED-modelBoth recognition families are able to handle multiple domains, and additionally, both are able to improve over the Transcript CRDNN systems trained on Finnish Parliament Train20. We find that limiting the language model data improves the language model perplexities on both Finnish Parliament and Lahjoita Puhetta, but in light of the WER results, the limiting is not helpful.

V. DISCUSSION

Our work highlights an open question: which speech recognition approach is the best one? We believe the Equal Data Setting and Matched Encoder Setting principles offer a compelling, fair alternative to competing for the state-of-the-art results. Even under the End-to-End Data limitation, and using the AED-model hyperparameters, the HMM/DNN-system consistently outperformed the AED-model in terms of WER in the CRDNN and wav2vec 2.0 experiments. In the Conformer experiments, with the hyperparameters tuned for the AED-model, and the HMM/DNN-system restricted to transcript language models and grapheme units, the AED-model did not surpass the HMM/DNN-system. One way to interpret these results is to see the HMM/DNN-system as a benchmark system: the results prove the room for improvement in the AED-model. Another interpretation is that although research focuses more and more on End-to-End speech recognition approaches, it is worthwhile to apply the neural network innovations to HMM/DNN-systems as well.

Our observations emphasize the need for more strictly controlled comparisons of heterogeneous speech recognition

TABLE XVII
ERROR RATES ON WORDS THAT DID NOT APPEAR IN THE TRAINING TRANSCRIPTS

Family	Encoder	Rare-WER [%]	
FP Train20		Test16	Test20
Occurences [num]		1029	1006
HMM FS HMM AED	CRDNN	29.15 44.51 35.76	45.92 56.56 48.41
HMM AED/CTC	w2v2.0	22.16 23.62	39.76 41.75
Librispeech		Clean	Other
Occurences [num]		330	437
HMM FS HMM AED	CRDNN	75.15 79.70 83.64	89.93 93.82 90.85
HMM AED	Conformer$_S^{FT}$ Conformer$_S^{MW}$	71.52 73.64	83.52 84.44
HMM FS HMM AED	Conformer$_L$ Conformer$_L^{FT}$ Conformer$_L^{MW}$	67.88 75.76 72.12	82.61 87.19 82.38
HMM AED/CTC	w2v2.0	61.82 66.36	75.06 77.35

The HMM/ DNN-systems use transcript language models.

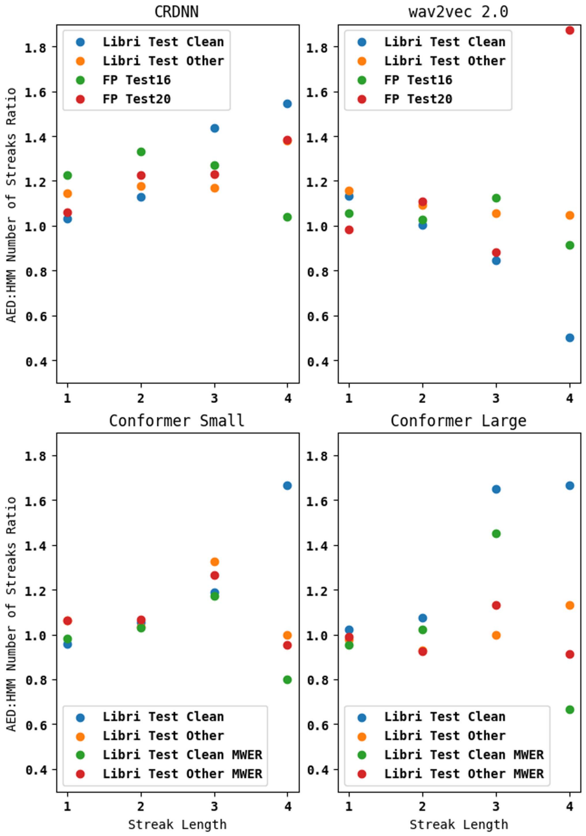


Fig. 1. Ratio of AED edit streaks per HMM edit streaks for streaks lengths 1 through 4.

systems in addition to competition for the state-of-the-art error rates. The thousand-hour scale is a particularly interesting ground for these comparisons, because no approach has proven conclusively better at that scale and because it is reachable in open datasets in many languages.

Analysing our empirical results, we find that the systems with similar performance still fail on different utterances - each have their own weaknesses. Between different HMM/DNN-systems, the systems using GMM alignments consistently outperform Flat Start systems. It appears that frame-level Cross Entropy training with GMM alignments is still useful for producing the best results, though we note that contrary results have also been presented [61]. However, we find that some simplifications of the HMM/DNN-system are possible, at least the thousand-hour scale. Tree-clustering for state-tying and phoneme-based units do not yield meaningful improvements in our experiments.

A. Limitations and Future Work

The experiments presented here leave some caveats regarding how the decoding-side implementations are matched. We believe something akin to a Matched Decoding Setting could be proposed in the future. This might match the language modeling context length, the use of neural language models in single-pass search and the language model capacity. Internal language model compensation (which matches the N-gram model probability replacement in neural language model rescoring) could also be a part of this.

Pure error rate performance is not the only relevant metric in choosing a speech recognition system. Our comparison does not consider for example the ability to deploy on mobile devices, the capability for online recognition (all encoder architectures use full utterance context in this work), nor the ease of development. We reported parameter counts, which matter for memory usage and model capacity, though in the latter area, neural network weights are not directly comparable with N-gram probabilities. If anything, the parameter counts probably favored the AED-models, which had in total more parameters than the corresponding HMM/DNN-system throughout all the experiments.

We sidestepped the favourite child problem by optimizing the AED-model and applying the hyperparamaters to the HMM/DNN-systems. This set an upper bound on the WER of the HMM/DNN-system. If the AED-model had outperformed the HMM/DNN-system, the conclusions would have been less clear. In that case, one solution could be to re-do the optimization in the other direction, applying the best HMM/DNN-system parameters to the AED-model.

Our practical experiments are naturally not able to cover all approaches. In particular, future work should include applying our proposed principles to comparisons involving Transducer models. Additionally, we concede that the manner in which we sidestepped the favourite child problem may lead to combinatorial amounts of work needed in comparisons involving more than two approaches. For example, had we attempted to include Transducers in this study, we would have had three pairs of approaches, with each pair potentially requiring their own set of hyperparameters and models.

Our analysis is able to show how models from different recognizer families make different errors, even though they may have similar performance. However, developing more advanced statistical methods would be a valuable contribution for the analysis of heterogeneous speech recognition systems.

VI. Conclusion

Choosing a speech recognition approach to use is currently difficult, both for deployment and for research, because there are many competing families of speech recognition approaches, each with their strengths and weaknesses.

We proposed two simple principles, and illustrated how those principles help to design more revealing speech recognition experiments. Experiments under the Equal Data Setting avoid confounding variables related to data, whereas experiments under the Matched Encoder Setting avoid confounding variables related to neural architecture and training. We demonstrated how to build AED-models and HMM/DNN-system adhering to these principles. During the course of developing our HMM/DNN-systems, we made multiple discoveries. We presented the multi-head decoding approach, and showed how GMM-alignments are still valuable for achieving the best results, though possibly not for tree-clustering.

We presented experiments on three thousand-hour-scale speech recognition tasks, comparing AED-models and HMM/DNN-systems. We optimize AED-models, reaching our HMM/DNN-system baselines from previous work. However, in comparisons under our proposed principles, our HMM/DNN-systems consistently yielded either equal or better error rates than AED-models. Our findings highlight the viability of HMM/DNN-systems in the era of End-to-End models.

Acknowledgment

The authors thank Felix McGregor for a discussion of the two-output-head HMM decoding method. We thank the kind anonymous reviewers for comments which improved this manuscript immensely. Computational resources were provided by Aalto ScienceIT and CSC – IT Center for Science, Finland.

References

[1] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 369–376.

[2] A. Graves, "Sequence transduction with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.: Representation Learn. Workshop*, 2012.

[3] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-Based Models for Speech Recognition," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 577—585.

[4] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2016, pp. 4960–4964.

[5] C.-C. Chiu et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 4774–4778.

[6] T. N. Sainath et al., "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 6059–6063.

[7] C. Lüscher et al., "RWTH ASR systems for LibriSpeech: Hybrid vs attention," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*2019, pp. 231–235, doi: 10.21437/Interspeech.2019-1780.

[8] A. Rouhe, A. Van Camp, M. Singh, H. Van Hamme, and M. Kurimo, "An equal data setting for attention-based encoder-decoder and HMM/DNN models: A case study in finnish ASR," in *Proc. 23rd Int. Conf. Speech Comput.*, 2021, pp. 602–613.

[9] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schlüter, and S. Watanabe, "End-to-end speech recognition: A survey," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 32, pp. 325–351, 2024, doi: 10.1109/TASLP.2023.3328283.

[10] M. Zeineldeen, J. Xu, C. Lüscher, R. Schlüter, and H. Ney, "Improving the training recipe for a robust conformer-based hybrid model," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 1036–1040.

[11] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015, pp. 5206–5210.

[12] R. Ardila et al., "Common voice: A massively-multilingual speech corpus," in *Proc. 12th Lang. Resour. Eval. Conf.*, 2020, pp. 4218–4222. [Online]. Available: https://aclanthology.org/2020.lrec-1.520

[13] A. Moisio et al., "Lahjoita puhetta: A large-scale corpus of spoken Finnish with some benchmarks," *Lang. Resour. Eval.*, vol. 57, pp. 1295–1327, 2022.

[14] D. Le, X. Zhang, W. Zheng, C. Fügen, G. Zweig, and M. L. Seltzer, "From senones to chenones: Tied context-dependent graphemes for hybrid speech recognition," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2019, pp. 457–464.

[15] T. Raissi, E. Beck, R. Schlüter, and H. Ney, "Improving factored hybrid HMM acoustic modeling without state tying," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 7442–7446.

[16] D. S. Park et al., "SpecAugment: A. simple data augmentation method for automatic speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 2613–2617.

[17] 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," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 551–555.

[18] Z. Tüske, G. Saon, and B. Kingsbury, "On the limit of english conversational speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 2062–2066.

[19] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2017, pp. 939–943.

[20] J. Li et al., "Recent advances in end-to-end automatic speech recognition," *APSIPA Trans. Signal Inf. Process.*, vol. 11, no. 1, 2022.

[21] P. Smit, S. Virpioja, and M. Kurimo, "Advances in subword-based HMM-DNN speech recognition across languages," *Comput. Speech Lang.*, vol. 66, 2021, Art. no. 101158. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885230820300917

[22] T. Hayashi et al., "Back-translation-style data augmentation for End-to-end ASR," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2018, pp. 426–433.

[23] N. Rossenbach, A. Zeyer, R. Schlüter, and H. Ney, "Generating synthetic audio data for attention-based speech recognition systems," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 7069–7073.

[24] M. Zeineldeen, A. Glushko, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, "Investigating methods to improve language model integration for attention-based encoder-decoder ASR models," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 2856–2860.

[25] M. Zeineldeen, A. Zeyer, W. Zhou, T. Ng, R. Schlüter, and H. Ney, "A systematic comparison of grapheme-based vs. phoneme-based label units for encoder-decoder-attention models," 2020, *arxiv/2005.09336*.

[26] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," in *Proc. Workshop Speech Natural Lang., Assoc. Comput. Linguistics*, 1992, pp. 357–362.

[27] J. S. Garofolo et al., "TIMIT acoustic phonetic continuous speech corpus," *Linguistic Data Consortium*, 1993, doi: 10.35111/17gk-bn40.

[28] S. Palaskar and F. Metze, "Acoustic-to-word recognition with sequence-to-sequence models," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2018, pp. 397–404.

[29] A. Zeyer, H. Ney, A. Merboldt, and R. Schlüter, "A comprehensive analysis on attention models," in *Proc. Interpretability Robustness Audio, Speech, Lang. Workshop*, 2018.

[30] W. Zhou, W. Michel, K. Irie, M. Kitza, R. Schlüter, and H. Ney, "The RWTH ASR system for TED-LIUM release 2: Improving hybrid hmm with specaugment," in *Proc. IEEE Int. Conf. Acoust.,Speech Signal Process.*, 2020, pp. 7839–7843.

[31] A. Rouhe, T. Kaseva, and M. Kurimo, "Speaker-aware training of attention-based end-to-end speech recognition using neural speaker embeddings," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 7064–7068.

[32] P. Doetsch, A. Zeyer, P. Voigtlaender, I. Kulikov, R. Schlüter, and H. Ney, "RETURNN: The RWTH extensible training framework for universal recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 5345–5349.

[33] Y. Wang et al., "Espresso: A fast End-to-end neural speech recognition toolkit," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2019, pp. 136–143.

[34] M. Ravanelli et al., "SpeechBrain: A general-purpose speech toolkit," 2021, *arXiv:2106.04624*.

[35] A. Gulati et al., "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 5036–5040, doi: 10.21437/Interspeech.2020-3015.

[36] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 12449–12460.

[37] D. Povey et al., "Purely sequence-trained neural networks for ASR based on Lattice-Free MMI," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2016, pp. 2751–2755.

[38] W. Reichl and W. Chou, "A unified approach of incorporating general features in decision tree based acoustic modeling," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1999, pp. 573–576.

[39] D. Povey et al., "The kaldi speech recognition toolkit," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2011, pp. 6465–6469.

[40] Y. Shao, Y. Wang, D. Povey, and S. Khudanpur, "PyChain: A fully parallelized PyTorch implementation of LF-MMI for end-to-end ASR," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 561–565.

[41] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2013, pp. 2345–2349.

[42] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, "End-to-end speech recognition using lattice-free MMI," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 12–16, doi: 10.21437/Interspeech.2018-1423.

[43] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 4694–4703.

[44] A. Rouhe, A. Virkkunen, J. Leinonen, and M. Kurimo, "Low resource comparison of attention-based and hybrid ASR exploiting wav2vec 2.0," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 3543–3547.

[45] T. Hirsimaki, J. Pylkkonen, and M. Kurimo, "Importance of high-order N-Gram models in morph-based speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 4, pp. 724–732, May 2009.

[46] V. Siivola, T. Hirsimaki, and S. Virpioja, "On growing and pruning Kneser–Ney smoothed N -gram models," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 5, pp. 1617–1624, Jul. 2007.

[47] E. Beck, R. Schlüter, and H. Ney, "LVCSR with transformer language models," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 1798–1802.

[48] 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. Acoust., Speech Signal Process.*, 2017, pp. 4835–4839.

[49] P. Sountsov and S. Sarawagi, "Length bias in encoder decoder models and a case for global conditioning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1516–1525. [Online]. Available: https://aclanthology.org/D16-1158

[50] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2017, pp. 523–527.

[51] R. Prabhavalkar et al., "Minimum word error rate training for attention-based sequence-to-sequence models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 4839–4843.

[52] A. Virkkunen, A. Rouhe, N. Phan, and M. Kurimo, "Finnish parliament ASR corpus," *Lang. Resour. Eval.*, vol. 57, pp. 1645–1670, 2023, doi: 10.1007/s10579-023-09650-7.

[53] M. Bisani and H. Ney, "Bootstrap estimates for confidence intervals in ASR performance evaluation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 2004, vol. 1, pp. 409.

[54] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[55] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *Proc. Conf. Uncertainty Artif. Intell.*, 2018, pp. 876–885. [Online]. Available: http://auai.org/uai2018/proceedings/papers/313.pdf

[56] Y. Yin et al., "Attention-based sequence-to-sequence model for speech recognition: Development of state-of-the-art system on Librispeech and its application to non-native english," 2018, *arxiv/1810.13088*.

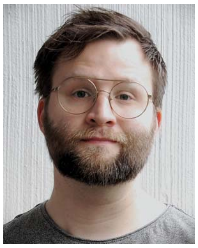
[57] K. Kim et al., "Attention based on-device streaming speech recognition with large speech corpus," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2019, pp. 956–963.

[58] K. Miyazaki, M. Murata, and T. Koriyama, "Structured state space decoder for speech recognition and synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.

[59] Y.-A. Chung et al., "W2V-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2021, pp. 244–250.

[60] M. G. Kendall, "The treatment of ties in ranking problems," *Biometrika*, vol. 33, no. 3, pp. 239–251, 1945. [Online]. Available: http://www.jstor.org/stable/2332303

[61] T. Raissi, C. Lüscher, M. Gunz, R. Schlüter, and H. Ney, "Competitive and resource efficient factored hybrid HMM systems are simpler than you think," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2023, pp. 4938–4942.



Aku Rouhe received the B.Sc. and M.Sc. degrees in 2014 and 2017, respectively, from Aalto University, Espoo, Finland, where he is currently working toward the Ph.D. degree. In 2020, he interned with Mila, Montreal, QC, Canada, and worked on the SpeechBrain toolkit. He is currently working at SiloGen on generative models. His research interests include speech recognition, and the on-going transition toward end-to-end models in machine learning.

Tamás Grósz received the Ph.D. degree in speech recognition from the University of Szeged, Szeged, Hungary, in 2018. Between 2017 and 2018, he was an Assistant Research Fellow with the Hungarian Academy of Sciences' Research Group on Artificial Intelligence. From 2018 to 2019, he was a Senior Lecturer with the Department of Computer Algorithms and Artificial Intelligence, University of Szeged. He is currently a Research Fellow with the Department of Information and Communications Engineering, Aalto University, Espoo, Finland. His research interests include automatic speech recognition, model explainability, and computational paralinguistics.

Mikko Kurimo (Senior Member, IEEE) received the D.Sc.Tech. degree in 1997. He is currently a Full Professor of speech and language processing and the Head of the Speech Recognition Group, Aalto University, Espoo, Finland. He has supervised 18 doctoral theses and 76 masters thesis and co-ordinated several large national (Academy of Finland and Business Finland) and international (EC and Nordforsk) research projects. He has authored or coauthored more than 240 peer reviewed international conference and journal publications. His research is focuses on machine learning in speech and language technology. His groups research results have been widely utilized via their open source tools and the achievements include, such as, winning the MGB-3 challenge for building low-resourced dialect ASR system and Compare 2022 stuttering and non-verbal vocalizations challenges and Compare 2023 spoken emotions challenge.