



Low Resource Comparison of Attention-based and Hybrid ASR Exploiting wav2vec 2.0

Aku Rouhe¹, Anja Virkkunen¹, Juho Leinonen¹, Mikko Kurimo¹

¹Aalto University, Department of Signal Processing and Acoustics, Finland

aku.rouhe@aalto.fi

Abstract

Low resource speech recognition can potentially benefit a lot from exploiting a pretrained model such as wav2vec 2.0. These pretrained models have learned useful representations in an unsupervised or self-supervised task, often leveraging a very large corpus of untranscribed speech. The pretrained models can then be used in various ways. In this work we compare two approaches which exploit wav2vec 2.0: an attention-based end-to-end model (AED), where the wav2vec 2.0 model is used in the model encoder, and a hybrid hidden Markov model (HMM/DNN) speech recognition system, where the wav2vec 2.0 model is used in the acoustic model. These approaches are compared in a very difficult Northern Sámi task, as well as an easier, simulated low resource task in Finnish. We find that the wav2vec 2.0 AED models can learn a working attention mechanism, but are still outperformed by wav2vec 2.0 HMM/DNN systems. Our best wav2vec 2.0 HMM/DNN recipe on 20 hours is competitive with an HMM/DNN system trained on 1600 hours.

Index Terms: speech recognition, low resource, wav2vec 2.0

1. Introduction

Most languages are low resource languages and will likely remain so in the foreseeable future. In contrast, state-of-the-art Automatic Speech Recognition (ASR) systems have primarily relied on large supervised corpora. However, the wav2vec family of models are spearheading a change in perspective: by leveraging very large unlabeled corpora in pretraining, only a small amount of supervised data is needed to build adequate ASR applications [1].

In parallel, another change of perspective is happening in ASR research: various end-to-end models, such as Connectionist Temporal Classification (CTC) [2], Recurrent Neural Network Transducer [3], and Attention-based Encoder-Decoder (AED) [4, 5] models have become competitive [6, 7] with Hybrid (as per the title) Hidden Markov Model / Deep Neural Network (HMM/DNN) systems. These changes have gone hand in hand: the most typical way to apply wav2vec models is in CTC models, but it has also been applied in AED models [8], and furthermore even in Cross-Entropy- and sequence discriminatively trained Hybrid HMM/DNN systems [9, 10].

In this work we compare how different approaches, AED models and HMM/DNN systems, are able to exploit wav2vec 2.0 in a low resource setting. The low resource setting sets novel challenges: even if the wav2vec 2.0 results have been impressive, can an AED model learn a meaningful attention mechanism from limited data? We also investigate if traditional Hidden Markov Model / Gaussian Mixture Model (HMM/GMM) based recipes can offer any benefit in a low resource scenario, taking the wav2vec 2.0 into account.

Our main findings are, **firstly**, that in an equal data low resource comparison of AED models and HMM/DNN systems powered by wav2vec 2.0, the HMM/DNN systems consistently outperform AED models. And **secondly**, how HMM/GMM recipes still consistently benefit the low resource HMM/DNN training, even if the HMM/GMM system itself does not perform well in large-vocabulary decoding.

2. Data

Two related Uralic languages are used: Northern Sámi and Finnish. Northern Sámi is a true low resource language: less than ten hours of transcribed speech were available to us, for example. Finnish is a high resource language, but we construct artificial low resource tasks in it. Both languages are agglutinative, as is typical of Uralic languages. Agglutinative languages lead to very large word vocabularies, a challenge which is sidestepped with subwords as language modeling units. Both languages have relatively transparent orthographies, which justifies the use of grapheme-based lexica.

2.1. Northern Sámi data

Two Northern Sámi corpora are used: *Giellagas North* [11] and the *UIT-SME TTS Corpus* (not publicly available). *Giellagas North* consists of interviews of Northern Sámi speakers - the speech is conversational and the recordings have varying amounts of noise. The interviewees speak three distinct dialects: Sea Sámi, Finnmark Sámi and Torne Sámi. The interviewers' speech is also transcribed, and is a part of the corpus. The *UIT-SME TTS Corpus* is designed for Text-To-Speech applications: the corpus contains clean, clear reading from two speakers, a man and a woman. The *UIT-SME TTS Corpus* is longer in duration, but *Giellagas North* has many more speakers. Both corpora have punctuation and capitalization - we normalize these away.

The *UIT-SME TTS Corpus* is distributed as long recordings without timestamps. As a preliminary processing step, an HMM/GMM system was bootstrapped on *Giellagas North* and finally trained using all Northern Sámi data. This system was only used to segment the *UIT-SME TTS Corpus*: utterance boundaries were placed at sentence ending punctuation.

The two Northern Sámi corpora are combined in a few different subsets and tasks to create two speaker dependent and two speaker independent tasks. Table 1 lists these tasks' details.

2.2. Finnish data

Just one Finnish corpus is used: the *Finnish Parliament ASR corpus* [12, 13]. This corpus is built of Parliament of Finland session recordings and their transcripts: the speech is mostly prepared. The corpus has two subsets. Here, specifically the *Train16* subset is used, because it has separate test sets for

speakers seen in the training data (Test-Seen) and speakers not seen in the training data (Test-Unseen). There are also corresponding validation sets (Dev-Seen and Dev-Unseen). A surprising, but replicated result is that the Unseen test sets have lower word error rates than the Seen counterparts.

Two different tasks are constructed: the Few-Speakers task, where the training and validation data only have speech from the eleven speakers that appear in Test-Seen, and an Many-Speakers task, where a random subset of Train16 utterances are picked. The Many-Speakers task uses the Dev-Unseen subset for validation. Both tasks have approximately the same number of utterances and are similar in duration, but the Many-Speakers task has many more speakers. Table 1 lists these tasks' details, and also describes the different test sets.

2.3. Language modeling data

Essentially all experiments use just the transcripts of the corresponding acoustic training data for language modeling. This has two benefits. Firstly, the end-to-end AED models and the HMM/DNN system can then be compared in an *equal data setting* [14]. Pure end-to-end models are only trained on transcribed speech. If the compared HMM-systems leverage large text-only corpora, this confounds differences in models and learning with differences in the training data. Secondly, this maintains a realistic, low resource setting. Although some general text data may be easier to come by than transcribed speech, good matching text data (e.g. conversational text) is more scarce.

3. Methods

The Wav2Vec 2.0 implementation is the same across all experiments in this work. It is exploited in various ways: as an Encoder in an AED model, or as an acoustic model in an HMM/DNN system.

3.1. wav2vec 2.0

The wav2vec family of models use self-supervised criteria to learn useful representations from large corpora of unlabeled speech. The wav2vec 2.0 [1] (W2V2) model solves a contrastive task over a codebook of quantized, jointly learned representations. The quantization module is discarded after learning. What remains is a convolutional front-end followed by a transformer context network. The standard wav2vec 2.0 Large architecture, which is used here, has 317M parameters.

Many different pretrained models are available online. All the wav2vec 2.0 experiments in this work base on the Uralic V2 model. It is trained on 42.5 thousand hours of three Uralic languages, Estonian, Finnish, and Hungarian, from the VoxPopuli corpus [15]. The Transformers library implementation¹ is used.

3.2. HMM/GMM systems

The HMM/GMM systems naturally do not leverage the wav2vec 2.0 models in any way. They are trained with a simple four stage (Kaldi-standard) recipe, and they all use a tri-state HMM topology with word-position dependent phones. Each stage uses all available training data, uses Mel-Frequency Cepstral Coefficient features, and is bootstrapped on alignment from the previous stage. Each triphone-GMM system targets

¹<https://huggingface.co/facebook/wav2vec2-large-uralic-voxpathuli-v2>

5000 Gaussians and uses tree-based state-tying with a maximum of 1200 leaves. The fourth, final (speaker-adaptively trained) HMM/GMM system results are reported in the results section. The systems are trained in the Kaldi toolkit [16].

3.3. HMM/DNN systems

The Hybrid HMM/DNN systems use the Kaldi Chain-style two-state HMM-topology and left-biphone state-tied tree [17]. The final HMM/GMM alignments are used in the state-tying algorithm and they also provide Cross-Entropy targets. The DNN acoustic model consists of the wav2vec 2.0 model and two fully connected layers. The acoustic model is trained with Chain-style multi-task setup of the Lattice-Free Maximum Mutual Information (LF-MMI) criterion and a separate output head using a Cross-Entropy criterion (10% weighted). The models are implemented in the SpeechBrain toolkit [18], with the LF-MMI implementation from PyChain [19].

Kaldi Chain-style models most typically use a one-per-30ms frame rate. However, the wav2vec 2.0 model has a one-per-20ms output frame rate, which does not neatly subdivide into one-per-30ms. In preliminary experiments, subsampling by two for a one-per-40ms frame rate lead to unstable training, and therefore, the wav2vec 2.0 native one-per-20ms output frame rate is used.

In inference, we use a novel approach where both DNN output heads are used: their outputs are passed through a LogSoftMax, and these log-likelihoods are interpolated linearly with the same weighting as used in training. In preliminary experiments we found that this yields a slight improvement over just using the LF-MMI output head. The log-likelihoods are fed to Kaldi lattice tools for decoding.

3.4. Flat Start HMM/DNN systems

We wished to investigate whether the alignments from the HMM/GMM systems offer any benefit in a low resource scenario. The PyChain LF-MMI criterion [19] can also be used for Flat Start (FS) training of HMM/DNN systems [20]. This uses a simple heuristically pruned biphone tree, and only uses the LF-MMI output head, and thus requires no alignments to start from. This system can leverage wav2vec 2.0 from the get-go. Otherwise the system is similar to that in the previous section.

3.5. HMM System Language Models

We use grapheme-based lexica and SentencePiece Byte Pair Encoding subword units [21] in language modeling. The lexicon transducer requires special word-position dependent phone handling [14, 22]. The language models are built with VariKN [23], which allows growing the modified Kneser-Ney models up to 10-gram scale. High-order n-grams are especially useful with subword units [24]. All our subword vocabularies use 400 BPE units.

3.6. Attention-based Encoder-Decoder models

All the AED models in this work are trained and applied end-to-end: they directly map speech to text. The encoder training is aided by an auxiliary Connectionist Temporal Classification criterion in a multi-task setup [25] in the initial epochs. The encoder consists of the wav2vec 2.0 model and two fully connected layers. The output of the wav2vec 2.0 model is subsampled by two in time, resulting in a typical AED encoder output frame rate of one-per-40ms. The decoder uses location-and-content-aware attention on the encoder outputs and consists of

Table 1: The tasks and their corresponding data subsets. Speaker overlap indicated for the results where parts or all speakers in validation and test sets appear in training data: shared letters indicate a set of shared speakers. A is UIT-SME. B and C is one way to divide Giellagas North, while another way is D, and E and F. X are the Finnish Parliament Test-Seen speakers, while Y and Z make up the rest of the training set. P and Q are the Dev-Unseen and Test-Unseen speakers, respectively.

	Number of Speakers	Speaker overlap	Number of Utterances	Size [hours]
All Northern Sámi Speaker Independent				
Train	7	A + D	5545	8.01
Validation	4	E	287	0.16
Test	10	F	1869	1.51
All Northern Sámi Speaker Dependent				
Train	21	A + B + C	6960	9.14
Validation	11	C	110	0.08
Test	11	C	631	0.48
Giellagas North Speaker Dependent				
Train	19	B + C	2046	1.57
Validation	11	C	110	0.08
Test	11	C	631	0.48
UIT-SME Speaker Independent Task				
Train	19	B + C	2046	1.57
Validation	11	C	110	0.08
Test	2	A	4914	7.57
Finnish Parliament Few-Speakers				
Train	11	X	6650	18.92
Validation	11	X	1537	4.46
Finnish Parliament Test-Seen	11	X	966	2.90
Finnish Parliament Many-Speakers				
Train	340	X + Y	6668	20.09
Validation	10	P	954	2.76
Finnish Parliament Test-Unseen	10	Q	962	2.81
Finnish Parliament Train16 Full	395	X + Y + Z	522 429	1559.45

Gated Recurrent Unit layers. The decoder outputs a distribution over the same BPE vocabulary as used by the corresponding HMM-system’s language model. In inference, beam search over the decoder output is used. The models are implemented in the SpeechBrain toolkit [18]

3.7. Connectionist Temporal Classification baseline

As an additional baseline, because of its popularity, we train CTC models for the Finnish tasks. These use the same architecture and training parameters as the HMM/DNN systems, except the output is a 400-unit BPE vocabulary. We do simple greedy decoding.

4. Experiments

HMM systems and AED models are trained on each different task and coefficients such as language model weight are chosen based on performance on the tasks’ validation data. All the wav2vec 2.0 subnetworks use a separate Adam optimizer with a 0.0001 learning rate, and the convolutional front-end parameters are not updated. The DNN acoustic models are trained for 50 nominal epochs of 500 updates, dynamically targeting 50 seconds of speech per minibatch. The DNN acoustic models use a learning rate of 0.1, whereas the AED models use a much smaller learning rate of 0.0001, and train for 50 nominal epochs of 1000 updates, with similar minibatching. All neural models in the Northern Sámi tasks leverage on-the-fly noise and reverberation augmentation, since the tasks were especially difficult. The Finnish models do not use augmentation.

The experiment implementations and further hyperparame-

ter choices can be found online².

5. Results

Table 2 lists the Finnish task results. Each task has its own corresponding validation set, but the results can be compared across the common test sets. The Test-Unseen set seems again to be easier than the Test-Seen set, as has been observed before [12].

Table 3 lists the results on the Northern Sámi tasks. The *All Northern Sámi Speaker Dependent* and *Giellagas North Speaker Dependent* tasks share the same validation and test sets, but otherwise results should only be compared within a task (divided by horizontal lines).

6. Discussion

The W2V2 HMM/DNN systems consistently outperform the W2V2 AED models, with larger relative performance gaps on the Finnish tasks. It is generally thought that the HMM systems have a more rigid structure of the ASR task built-in, which helps them perform better in low resource scenarios; the results here appear to support that conclusion. However, it is remarkable that the W2V2 AED models are able to learn an attention mechanism at all, and one which even generalizes to unseen speakers, on less than 20 hours of data from 11 speakers. It is noteworthy that the W2V2 AED model gains much more from seeing a large variety of speakers, than the W2V2 HMM/DNN system does, on the FP Unseen-Test data (comparing from Few-

²<https://github.com/aalto-speech/kaldi-sb-north-sme>

Table 2: Results for Finnish on task-specific validation sets and the Finnish Parliament (FP) Test-Seen and Test-Unseen sets. Both Word Error Rate (WER) and Character Error Rate (CER) results are shown - models are optimized for WER.

	WER/CER [%]		
	Valid (Corresp.)	FP Test-Seen	FP Test-Unseen
Finnish Parliament Few-Speakers			
W2V2 AED	24.11 / 13.53	19.47 / 9.44	20.33 / 9.61
W2V2 HMM/DNN	15.37 / 8.44	11.86 / 4.19	11.11 / 3.75
W2V2 FS HMM/DNN	20.46 / 8.13	18.25 / 4.07	17.87 / 3.65
W2V2 CTC _{greedy}	24.86 / 10.73	19.92 / 5.89	18.43 / 5.35
HMM/GMM	44.86 / 17.65	40.52 / 11.90	40.13 / 11.61
Finnish Parliament Many-Speakers			
W2V2 AED	16.54 / 6.75	15.74 / 6.98	13.88 / 6.11
W2V2 HMM/DNN	13.93 / 4.93	11.92 / 3.82	10.13 / 3.34
W2V2 FS HMM/DNN	20.14 / 5.10	17.79 / 4.04	16.85 / 3.54
W2V2 CTC _{greedy}	20.89 / 6.76	18.86 / 5.68	17.58 / 5.19
HMM/GMM	43.79 / 14.12	40.75 / 12.26	37.32 / 11.13
Finnish Parliament Train16 Full			
AED [13]	-	12.60 / 5.89	11.95 / 5.48
HMM/DNN [13]	-	11.46 / 4.30	10.86 / 4.07

Table 3: Results for Northern Sámi tasks.

	WER/CER [%]	
	Valid	Test
All Northern Sámi Speaker Independent		
W2V2 AED	81.06 / 50.53	71.35 / 39.04
W2V2 HMM/DNN	79.20 / 56.76	72.66 / 46.53
HMM/GMM	99.72 / 98.74	96.64 / 89.94
All Northern Sámi Speaker Dependent		
W2V2 AED	48.62 / 23.46	53.10 / 26.67
W2V2 HMM/DNN	45.85 / 25.21	51.78 / 29.65
W2V2 FS HMM/DNN	52.36 / 26.79	57.26 / 29.35
HMM/GMM	81.63 / 60.22	83.25 / 58.79
Giellagas North Speaker Dependent		
W2V2 AED	58.37 / 33.52	63.26 / 35.58
W2V2 HMM/DNN	55.61 / 27.83	57.29 / 28.54
W2V2 FS HMM/DNN	59.51 / 38.99	64.94 / 41.47
HMM/GMM	76.59 / 48.40	81.64 / 53.33
UIT-SME Speaker Independent Task		
W2V2 AED	-	79.65 / 43.01
W2V2 HMM/DNN	-	63.17 / 21.48
W2V2 FS HMM/DNN	-	71.28 / 35.50

Speakers to Many-Speakers tasks). The CTC models are presented for reference, and conclusions should not be drawn from them.

Both the speaker independent and even the speaker dependent ASR performance on Northern Sámi is still generally inadequate for most applications. The Finnish models with a similarly small number of speakers generalize to unseen speakers. However, the speaker independent Northern Sámi tasks have much worse error rates, meaning that the Sámi models do suffer from missing speaker variety. Thus we believe there are four factors at play simultaneously: 1) the Sámi datasets are too small, 2) with too few speakers, 3) consist of difficult conversational and dialectal speech, and 4) Sámi is not included in the wav2vec 2.0 pretraining data. Even modest increases in gathered data, particularly from new speakers, might be enough to bring the error rates down drastically, judging based on the Finnish results. However, if the lack of Sámi in the pretraining data is a key issue, gathering thousands of hours of unlabeled

speech could unfortunately be difficult - a setback for the hopes raised by the wav2vec family. Fortunately, it is at least known that on the cleaner and clearer UIT-SME TTS data, speaker dependent models can already work well [26].

The Flat Start W2V2 HMM/DNN models consistently fare worse than their HMM/GMM-alignment-powered W2V2 HMM/DNN counterparts in terms of word error rate. As a curious result, the Flat Start models show great promise in the CER results. Although the HMM/GMM systems do not work in large-vocabulary decoding at this data scale, they still seem to be valuable in tree building and for Cross-Entropy targets. Impressively, the W2V2 HMM/DNN trained on less than 20 hours of data from 11 speakers is practically on-par with models trained on 1600 hours - and this difference in training data is mirrored in the (transcript-only) language modeling data.

The results here show that both AED models and HMM/DNN systems benefit meaningfully from pretraining. A similar comparison on self-training is a natural future direction, as it is a complementary method [27].

7. Conclusions

We compared AED models as well as HMM/DNN systems and Flat Start HMM/DNN systems, all exploiting wav2vec 2.0 pretraining, in various low resource tasks. In general the Northern Sámi tasks were difficult, with any meaningful results coming from speaker dependent scenarios. The AED models generally performed worse than the HMM/DNN systems, but learning an attention mechanism on such low resource tasks at all is promising. The Flat Start HMM/DNN systems fared generally worse than HMM/DNN systems which leveraged HMM/GMM-alignments, showing that HMM/GMM systems can still be valuable. Our recipes offer templates of how ASR systems can be built for low resource languages by exploiting wav2vec 2.0.

8. Acknowledgements

We are grateful for the Academy of Finland project funding, numbers: 337073, 345790. We acknowledge the computational resources provided by the Aalto Science-IT project.

9. References

- [1] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 12 449–12 460.
- [2] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.
- [3] A. Graves, A. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645–6649.
- [4] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, “End-to-end attention-based large vocabulary speech recognition,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 4945–4949.
- [5] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.
- [6] C. Lüscher, E. Beck, K. Irie, M. Kitzka, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, “RWTH ASR Systems for LibriSpeech: Hybrid vs Attention,” in *Proc. Interspeech 2019*, 2019, pp. 231–235. [Online]. Available: <http://dx.doi.org/10.21437/Interspeech.2019-1780>
- [7] C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, “State-of-the-art speech recognition with sequence-to-sequence models,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4774–4778.
- [8] C. Yi, J. Wang, N. Cheng, S. Zhou, and B. Xu, “Transfer ability of monolingual Wav2vec2.0 for low-resource speech recognition,” in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–6.
- [9] A. Vyas, S. Madikeri, and H. Bourlard, “Lattice-free mmi adaptation of self-supervised pretrained acoustic models,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6219–6223.
- [10] J. Zhao, G. Shi, G.-B. Wang, and W.-Q. Zhang, “Automatic speech recognition for low-resource languages: The three systems for the IARPA Openasr20 evaluation,” in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 335–341.
- [11] “Pohjoissaamen näytekorpus.” [Online]. Available: <http://urn.fi/urn:nbn:fi:lb-201407302>
- [12] A. Mansikkaniemi, P. Smit, and M. Kurimo, “Automatic construction of the Finnish Parliament speech corpus,” in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. International Speech Communication Association, Aug. 2017, pp. 3762–3766.
- [13] A. Virkkunen, A. Rouhe, N. Phan, and M. Kurimo, “Finnish Parliament ASR corpus - Analysis, benchmarks and statistics,” 2022.
- [14] 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 *Speech and Computer*, A. Karpov and R. Potapova, Eds. Cham: Springer International Publishing, 2021, pp. 602–613.
- [15] C. Wang, M. Riviere, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino, and E. Dupoux, “VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 993–1003. [Online]. Available: <https://aclanthology.org/2021.acl-long.80>
- [16] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, “The Kaldi Speech Recognition Toolkit,” in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, 2011.
- [17] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, “Purely sequence-trained neural networks for ASR based on Lattice-Free MMI,” in *Interspeech 2016*, 2016, pp. 2751–2755.
- [18] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, “Speechbrain: A general-purpose speech toolkit,” 2021.
- [19] 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. Interspeech 2020*, 2020, pp. 561–565.
- [20] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, “End-to-end Speech Recognition Using Lattice-free MMI,” in *Proc. Interspeech 2018*, 2018, pp. 12–16.
- [21] T. Kudo and J. Richardson, “SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71.
- [22] P. Smit, S. Virpioja, and M. Kurimo, “Improved Subword Modeling for WFST-Based Speech Recognition,” in *Proc. Interspeech 2017*, 2017, pp. 2551–2555.
- [23] V. Siivola, M. Creutz, and M. Kurimo, “Morfessor and VariKN machine learning tools for speech and language technology,” in *8th Annual Conference of the International Speech Communication Association (Interspeech 2007)*, Antwerp, Belgium, August 27-31, 2007. ISCA, 2007, pp. 1549–1552.
- [24] T. Hirsimäki, J. Pytkkonen, and M. Kurimo, “Importance of high-order n-gram models in morph-based speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, pp. 724–732, 2009.
- [25] S. Kim, T. Hori, and S. Watanabe, “Joint CTC-attention based end-to-end speech recognition using multi-task learning,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 4835–4839.
- [26] J. Leinonen, P. Smit, S. Virpioja, and M. Kurimo, “New baseline in automatic speech recognition for Northern Sámi,” in *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages*. Helsinki, Finland: Association for Computational Linguistics, Jan. 2018, pp. 87–97. [Online]. Available: <https://aclanthology.org/W18-0208>
- [27] Q. Xu, A. Baevski, T. Likhomanenko, P. Tomasello, A. Conneau, R. Collobert, G. Synnaeve, and M. Auli, “Self-training and pre-training are complementary for speech recognition,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 3030–3034.