INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Combining Hybrid and End-to-end Approaches for the OpenASR20 Challenge

Tanel Alumäe, Jiaming Kong

Tallinn University of Technology, Estonia
tanel.alumae@taltech.ee, kinetical@live.com

Abstract

This paper describes the TalTech team submission to the OpenASR20 Challenge. OpenASR20 evaluated low-resource speech recognition technologies across 10 languages, using only 10 hours of training data in the constrained condition. Our ASR systems used hybrid CNN-TDNNF-based acoustic models, trained with different data augmentation strategies. We used language model adaptation, recurrent neural network language models and lattice combination for improving first pass results. The scores of our submissions were the best across all teams in six out of ten languages. The paper also describes post-evaluation experiments that focused on the unconstrained condition. We show that optimized N-best list combination of a CNN-TDNNF based system and a finetuned multilingual XLSR-53 model results in large reductions in word error rate. Using BABEL data and the combination of hybrid and end-to-end systems gives 12-22% relative improvement over the constrained condition results.

Index Terms: OpenASR20, speech recognition, low-resource, wav2vec, XLSR-53

1. Introduction

The OpenASR (Open Automatic Speech Recognition) Challenge is organized by NIST with the goal of evaluating speech recognition technologies under low resource constraints.

The 2020 edition of the challenge (OpenASR20) [1] consisted of speech recognition tasks for 10 languages. The audio data consisted of conversational telephone speech. There were two training conditions for all languages: constrained and unconstrained. The constrained condition restricted the acoustic training data only to the provided 10 hour subset of the Build dataset. Additional text data from any public sources may be used for training. The unconstrained training condition allowed using additional publicly available speech and text training data from any language for training the models.

Low-resource speech recognition as well as keyword spotting has been a popular research topic in the last decade. Both hybrid and end-to-end approaches were investigated within the IARPA BABEL program (e.g. [2, 3, 4, 5]). Extremely low resource end-to-end speech recognition has been studied in recent years from the perspective of transfer learning. In [6] it was shown that self-supervised representations trained on large amounts of untranscribed speech enable to train an accurate speech recognition model on just 10 minutes of labeled speech. In [7], this approach was extended to multi-lingual setting. An alternative approach is to perform multilingual low-resource speech recognition using a single model [8], or to use meta-learning for more efficient transfer from a multilingual model to several low-resource languages [9, 10].

The Tallinn University of Technology (TalTech) team participated in the constrained training condition of the OpenASR Challenge for all 10 languages. The scores of our submissions were the best across all teams in six out of ten languages.

Table 1: *IARPA BABEL language packs used for additional textual training data and post-evaluation experiments, together with the corresponding amount of conversational speech in hours.*

Language	Language pack	LDC ID	#h
Amharic	IARPA-babel307b-v1.0b	LDC2019S22	44
Cantonese	IARPA-babel101b-v0.4c	LDC2016S02	141
Guarani	IARPA-babel305b-v1.0c	LDC2019S08	43
Javanese	IARPA-babel402b-v1.0b	LDC2020S07	46
K-Kurdish	IARPA-babel205b-v1.0a	LDC2017S22	42
Mongolian	IARPA-babel401b-v2.0b	LDC2020S10	46
Pashto	IARPA-babel104b-v0.4bY	LDC2016S09	78
Tamil	IARPA-babel204b-v1.1b	LDC2017S13	69
Vietnamese	IARPA-babel107b-v0.7	LDC2017S01	87

This paper gives a detailed description of our approach that is based on hybrid HMM-DNN acoustic models, language model adaptation, lattice rescoring using recurent neural network language models and lattice combination. In addition, the paper provides a post-evaluation analysis for both constrained and unconstrained conditions. We show that modern end-to-end models based on the Conformer [11] architecture perform very poorly with only 10 hours of training data. However, finetuning a pretrained multilingual XLSR-53 wav2vec 2.0 model [7] on the constrained data results in word error rates (WER) that are lower than the well-tuned combination of hybrid DNN-HMM models for several languages. We show that the fusion of hybrid and end-to-end systems via optimized combination of N-best lists results in solid WER reductions. Finally, we provide post-evaluation results in the unconstrained condition for five languages, based on a fusion of hybrid and finetuned XLSR-53 models, resulting in an average relative improvement of 20% over the constrained system WER.

2. Description of the TalTech submission

2.1. Training data

Since we participated in the Constrained training condition, the only acoustic data that we used was the 10-hour subset of the Build dataset provided for each language.

For language modeling, we used additional training data for all languages. For all languages except Somali, we use the speech transcripts from IARPA BABEL language packs as additional textual training data (see Table 1), together with the pronunciation lexicons in the language packs.

Since there is no BABEL language pack for Somali, we used the Somali Web Corpus [12] as additional textual training data for this language. We used the corpus only to extend the language model vocabulary by 500 000 most frequent words. Using the corpus for training the actual language models did not improve language model performance.

Copyright © 2021 ISCA http://dx.doi.org/10.21437/Interspeech.2021-1086

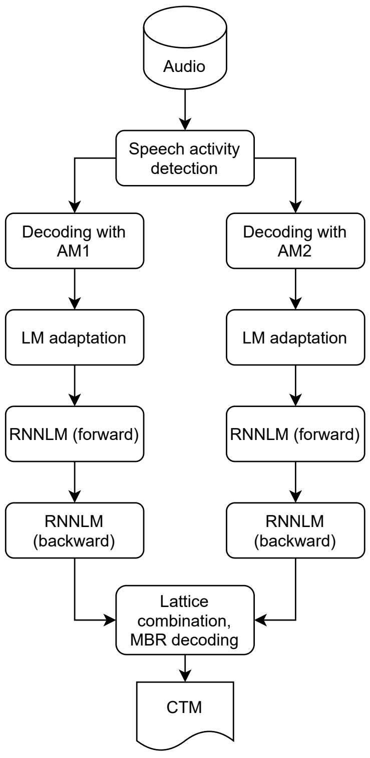

Figure 1: *Workflow of the decoding pipeline.*

2.2. Decoding pipeline

Figure 1 gives a visual representation of the decoding pipeline. Some certain rescoring steps are omitted for some of the languages (see below).

2.3. Acoustic Modelling

Our systems are based on the hybrid DNN-HMM approach using Kaldi [13] as the main software tool. We trained two acoustic models with identical architecture for each language. The models use the CNN-TDNNF topolology [14] and are trained according to the Kaldi "chain" model training approach [15]. There are three input feature streams: 40-dimensional filterbank features, 100-dimensional i-vectors (updated every 100 milliseconds) and 3-dimensional pitch features. The i-vector and pitch features are transformed into spatial 40-dimensional planes (five for i-vector features, one for pitch features) using learned linear layers and are combined with the filterbank features. The resulting seven $40 \times T$ planes (where T corresponds to the time dimension) are first processed using six 3×3 convolutional layers, each using the ReLU activation function. The output from the convolutional block is processed by nine TDNNF layers.

For each language, we trained two acoustic models: one on clean speed-perturbed data and the other on noise-augmented data. SpecAugment was used during the training of the first model, and the model was trained for 20 epochs. For noise augmentation, we used the standard multi-condition training approach implemented in Kaldi [16]: four copies were made from the clean speed-perturbed data, and the copies were reverberated, mixed with background noise, music, or babble noise, respectively. Noises from the MUSAN corpus [17] were used for augmentation. Since there was now four times more training data, the second model was trained for only five epochs.

For the nine languages that had IARPA BABEL language packs, the pronunciation lexicons are produced from the corresponding language pack lexicons. We used the provided lexicons with the following minor changes:

- Amharic: all the labialized consonants were split into two: the main phoneme and a labial pseudo-phoneme (e.g., $p^w \to p\ w$);
- Kurmanji-Kurdish: we removed the distinction between phonemes in stressed and unstressed syllables.

For Somali, we used the provided pronunciation lexicon in the Build dataset for training a grapheme-to-phoneme model using Phonetisaurus [18] which was then used for generating pronunciations for words from the external text corpus.

2.4. Language Modelling

The language model used for decoding is a maximum entropy 4-gram model trained using SRILM [19, 20].

The lattices from the first decoding pass are rescored using language models adapted to the given conversation side. This is done similarly to the method described in [21]. The language model training data is divided into "documents", representing one conversation side. Then, the hypotheses from the first decoding pass are used to find conversations in the training data that are most useful for increasing the unigram perplexity of the first pass hypotheses of the individual conversations. In other words, we look for such documents in the training corpus that produce a language model that improve perplexity of the conversation transcripts, when applied in interpolation with the background model. For each conversation side being processed, we select all such documents from the training corpus that increase the perplexity of the first pass transcripts. The set of selected documents is then used for adapting the "background" unadapted maximum entropy 4-gram language model for each conversation side. The adaptation is performed as described in [22]: during optimization of the parameters for a certain conversation, the parent model was taken as a prior. This method encourages the domain-specific models to have feature weights close to the prior model using regularization, if there is little evidence to change them. This was done using the SRILM extension for maximum entropy language models [20].

The lattices that are rescored using adapted language models are further rescored using two recurrent neural network language models (RNNLMs), one running in forward direction and the other running in backward direction. The RNNLMs are trained using Kaldi [23] and consist of two LSTM layers with the cell dimensionality of 200. Word embedding dimensionality is also 200. The so-called Backstitch optimization method [24] was used for training RNNLMs, with the exception of some languages (Mongolian and Vietnamese) for which Backstitch caused the RNNLMs not to converge. For those languages, RNNLMs were trained without Backstitch. For Amharic, the RNNLMs did not improve the recognition results at all on development data and we didn't use them for rescoring evaluation data.

2.5. Speech Activity Detection

Since the evaluation data is not segmented into utterances, we trained a speech activity detection model to detect regions of speech in the test data, using the implementation in Kaldi. This approach first trains a GMM speech recognition model on the provided clean training data, and then combines the labels from

Table 2: *Final WER (%) results for different languages in the Constrained condition. The best WER of the challenge over all teams is given in the last column. For six languages, marked in bold, our submission achieved the lowest WER.*

Language	Dev	Eval	Best Eval
Amharic	37.0	**45.1**	45.1
Cantonese	47.2	45.4	40.2
Guarani	40.3	46.6	46.1
Javanese	53.7	53.8	52.1
Kurmanji-Kurdish	63.5	**65.3**	65.3
Mongolian	48.0	47.3	45.4
Pashto	43.6	**45.7**	45.7
Somali	57.1	**59.1**	59.1
Tamil	62.5	**65.1**	65.1
Vietnamese	45.2	**45.1**	45.1

the alignment with the GMM model with default non-speech labels for unlabeled regions of the training data. The resulting training data is augmented with reverberation and noise perturbation, and the final TDNN-based speech activity detection model is trained. The model uses statistics pooling for incorporating long-range information.

2.6. Results

Table 3: *Word error rates (%) for on Guarani and Javanese development sets after individual decoding phases.*

	Guarani		Javanese	
Acoustic model	Spec-Augment	Multi-condition	Spec-Augment	Multi-condition
1st pass	44.0	44.1	56.2	56.1
+ LM adaptation	43.6	43.9	55.7	55.7
+ RNNLM (fwd)	42.1	42.3	54.9	54.9
+ RNNLM (bwd)	41.8	41.9	55.2	55.2
Combination	40.3		53.7	

Table 2 lists WERs for development and evaluation sets across all languages. Development set results are taken from our internal decoding and scoring runs, while the results on the evaluation data originate from the official leaderboards. In six out of ten languages (marked in bold), our submissions scored the best WER across all submissions.

Table 3 shows the WERs after each decoding step on the development sets of two languages, Gurarani and Javanese. For most languages, the trend was similar to those two languages: first pass decoding using either of the two acoustic models resulted in similar WERs, although the absolute WERs across languages were very different. Lattice rescoring and combination resulted in 5-10% relative improvement, with regard to the first pass results.

3. Post-evaluation experiments

3.1. Comparison of hybrid, end-to-end and pretrained models on the constrained data

In the post-evaluation phase, we compared the performance of the hybrid model that we used in the OpenASR20 submission with two end-to-end approaches when using only the provided 10 hour training dataset. First, we experimented with the Conformer model [11]. The Conformer model utilizes the convolution-augmented transformer. This structure captures both the global dependencies and local feature patterns to achieve state-of-the-art accuracies on several benchmark datasets, while also improving parameter efficiency. In our experiment, we trained a Conformer model whose structure was identical to the original paper [11], and a Tranformer language model as shown in [25] on the BABEL training data transcripts. The only input to the Conformer model were the 40-dimensional filterbank features. SpecAugment was also used in the training of Conformer model. With warmup steps, the Conformer model for each language was trained for 80 epochs, and the Transformer LM was trained for 50 epochs.

Table 4: *Comparison of different model architecture performances on the development data of five languages, using only the 10h Build dataset for training/finetuning. Note that XLSR-53 was exposed to unlabeled Cantonese and Kurmanji-Kurdish BABEL data during pretraining.*

	Amharic	Cantonese	Javanese	Kurm.-Kurdish	Mongolian
CNN-TDNNF	38.1	49.9	55.3	65.4	51.2
Conformer	94.6	99.4	95.2	95.8	90.6
Finet. XLSR-53	42.8	43.6*	51.3	62.0*	46.6

As an alternative end-to-end approach, we experimented with finetuning the publicly available XLSR-53 wav2vec 2.0 model [7]. XLSR-53 is a large pretrained model trained on unlabeled multilingual data. The model is trained by jointly solving a contrastive task over masked latent speech representations and learning a quantization of the latents shared across languages. The model contains a convolutional feature encoder that maps raw audio to latent speech representations which are fed to a Transformer network that outputs context representations. XLSR-53 is pretrained on 56 000 hours of speech data from 53 languages. Pretraining data includes 650 hours of BABEL training data from 10 languages (Bengali, Canotonese, Kazakh, Haitian, Kurmanji-Kurdish, Pashto, Tamil, Turkish, Tok Pisin and Vietnamese). Finetuning the model to a particular language is performed by adding a classifier representing the output vocabulary on top of the pretrained model and training on the labeled data with the Connectionist Temporal Classification (CTC) loss [26]. For this experiment, we finetuned XLSR-53 on the OpenASR20 10h Build dataset of the particular language. We used configuration given in the wav2vec2.0 code repository[1] that finetunes the model for 20k updates using 4 GPUs. Graphemes were used in the output vocabulary. Decoding was performed using a 4-gram language model estimated on the BABEL training data transcripts. Note that using a model pretrained on external audio data is not allowed by the rules of the constrained condition. We still include the results of this experiment in this section as we believe that using such pretrained model is a highly practical choice for a realistic scenario when only 10 h of training data is available.

Table 4 lists WERs on development data of five languages. The DNN-HMM model corresponds to the CNN-TDNNF model trained with SpecAugment for our submission. The results are based on reference speech segments, hence the slight differences with Table 3. It can be seen that training a

[1] `https://github.com/pytorch/fairseq/blob/master/examples/wav2vec/config/finetuning/base_10h.yaml`

Table 5: *WERs of various unconstrained systems built in the post-evaluation phase, and their comparison to the best OpenASR20 systems of the unconstrained condition.*

		Amharic		Javanese		Mongolian		Cantonese		Kurmanji-Kurdish	
	Amount of BABEL data	44 h		46 h		46 h		141 h		42 h	
ID	System	Dev	Eval	Dev	Eval	Dev	Eval	Dev	Eval	Dev	Eval
A	Our official submission (constrained cond.)	37.0	45.1	53.7	53.8	48.0	47.3	47.2	45.4	63.5	65.3
B	XLSR-53, finetuned on 10h	43.0		52.5		47.8		45.7		63.1	
C	Fusion A+B	35.1		49.9		44.6		43.1		60.6	
D	CNN-TDNNF + RNNLM, on BABEL data	32.7		49.2		42.9		42.0		59.7	
E	XLSR-53, finetuned on BABEL data	34.3		45.9		38.9		39.3		57.8	
F	Fusion D+E	28.9	**35.2**	44.0	**44.1**	36.0	**36.9**	38.0	37.4	55.7	**57.7**
	Best OpenASR20 unconstrained system		-		-		40.6		32.0		-
	Relative improvement of F w.r.t. A		22.0%		18.0%		22.0%		17.6%		11.6%

state-of-the-art Conformer model from scratch only on the 10h data does not produce a useful model. However, the results of a hybrid CNN-TDNNF model and the finetuned XLSR-53 model are relatively similar. XLSR-53 has the best WER for four out of five languages. It must be emphasized that the comparison is not totally fair, as XLSR-53 was exposed to unlabeled Cantonese and Kurmanji-Kurdish language during pretraining.

3.2. Experiments with unconstrained data

Our second set of post-evaluation experiments focused on the unconstrained condition. In order to save computational resources, we performed the experiments on five out of ten OpenASR20 languages. For all languages, we used the full BABEL corpus as additional training data, containing around 40 to 140 hours of transcribed speech.

The results are given in Table 5. System A corresponds to our official submission and system B to the finetuned XLSR-53 model described in the previous section, when using speech segments generated using SAD.

We trained two additional systems on the BABEL data for each language. The hybrid CNN-TDNNF system (D in Table 5) is similar to the system used in our submission for the constrained condition, with the exception that it uses only one larger acoustic model (dimension of TDNNF layers is increased from 768 to 1024), trained on the BABEL data. Language model adaptation and lattice rescoring with forward and backward RNNLMs are also applied. We also finetuned XLSR-53 on full BABEL data (system E in Table 5). Finetuning was performed similarly as with the 10h training data, except that the model was trained for more updates (5000 instead of 2000) and using larger learning rate. It can be seen that the finetuned XLSR-53 model achieves lower WER scores than the hybrid system for four out of five languages.

Finally, we experimented with combining hybrid and end-to-end systems. Although it is well known that combining systems using different architecture and modelling approaches often results in large improvements, combining hybrid and end-to-end models efficiently is not straightforward. Combining the CTM files of the individual systems using ROVER does not result in substantial improvements, mostly because the words generated by the end-to-end system typically lack confidence scores. We adopt the N-best list combination method that is similar to the method proposed in [27], consisting of the following steps and implemented using SRILM [19]:

1. N-best (we used $N = 50$) lists are generated for both systems, including acoustic and language model scores for each hypothesis;
2. Optimized weights for acoustic and language model scores and sentence length normalization factors are found for each system separately, using simplex-based "Amoeba" search of the lowest WER on development data;
3. N-best list combination weights are optimized on the development data, using the resulting posteriors of reference words as the optimization target;
4. N-best lists with optimized score combination weights are decoded using word error minimization, in a generalization of the ROVER algorithm [28].

Table 5 shows that the fusion of hybrid and XLSR-53 based systems (C and F) gives consistent improvements across languages, regardless of which of the source systems was stronger. The fusion of systems trained on unconstrained data (F) results in WERs that are the best reported numbers in the OpenASR20 unconstrained condition for four languages out of five (at the time of writing this paper). Using additional data and a fusion with the finetuned XLSR-53 model gives 12-22% relative improvement with regard to the well-tuned system built only using 10h of speech data.

4. Conclusion

In this paper, we presented the TalTech submission to the OpenASR20 Challenge that achieved the top score in six out of ten languages in the constrained condition. The paper also described our post-evaluation experiments in the unconstrained condition. We showed that finetuning the XLSR-53 wav2vec 2.0 model is a viable alternative to hybrid Kaldi-based systems with both 10h and 40-140 hours of training data, and the N-best list based combination gives substantial improvements over either of the single systems. In the future, we plan to experiment with multilingual pretraining of the hybrid system and combine XLSR-53 based feature extractor with the state-of-the-art Conformer model.

5. Acknowledgements

The work was partly done within the project "ICT programme", supported by the European Union through the European Social Fund. The authors acknowledge the TalTech supercomputing resources made available for conducting the research reported in this paper.

6. References

[1] "OpenASR20 challenge evaluation plan," https://www.nist.gov/system/files/documents/2020/10/21/OpenASR20_EvalPlan_v1_5.pdf, accessed: 2020-11-23.

[2] A. Rosenberg, K. Audhkhasi, A. Sethy, B. Ramabhadran, and M. Picheny, "End-to-end speech recognition and keyword search on low-resource languages," in *ICASSP*, 2017.

[3] W. Hartmann, D. Karakos, R. Hsiao, L. Zhang, T. Alumäe, S. Tsakalidis, and R. Schwartz, "Analysis of keyword spotting performance across IARPA BABEL languages," in *ICASSP*, 2017.

[4] M. J. Gales, K. M. Knill, and A. Ragni, "Low-resource speech recognition and keyword-spotting," in *SPECOM*, 2017.

[5] T. Alumäe, S. Tsakalidis, and R. M. Schwartz, "Improved multilingual training of stacked neural network acoustic models for low resource languages." in *Interspeech*, 2016.

[6] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[7] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *arXiv preprint arXiv:2006.13979*, 2020.

[8] S. Zhou, S. Xu, and B. Xu, "Multilingual end-to-end speech recognition with a single transformer on low-resource languages," *arXiv preprint arXiv:1806.05059*, 2018.

[9] J.-Y. Hsu, Y.-J. Chen, and H.-y. Lee, "Meta learning for end-to-end low-resource speech recognition," in *ICASSP*, 2020.

[10] W. Hou, H. Zhu, Y. Wang, J. Wang, T. Qin, R. Xu, and T. Shinozaki, "Exploiting adapters for cross-lingual low-resource speech recognition," *arXiv preprint arXiv:2105.11905*, 2021.

[11] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Interspeech*, 2020.

[12] V. Suchomel and P. Rychlý, "Somali web corpus," 2016, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. [Online]. Available: http://hdl.handle.net/11234/1-2591

[13] 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 *ASRU*, 2011.

[14] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," *Interspeech*, 2018.

[15] H. Hadian, D. Povey, H. Sameti, J. Trmal, and S. Khudanpur, "Improving LF-MMI using unconstrained supervisions for ASR," in *SLT*, 2018.

[16] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *ICASSP*, 2017, pp. 5220–5224.

[17] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[18] J. R. Novak, N. Minematsu, and K. Hirose, "Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the WFST framework," *Natural Language Engineering*, vol. 22, no. 6, pp. 907–938, 2016.

[19] A. Stolcke, J. Zheng, W. Wang, and V. Abrash, "SRILM at sixteen: Update and outlook," in *ASRU*, vol. 5, 2011.

[20] T. Alumäe and M. Kurimo, "Efficient estimation of maximum entropy language models with n-gram features: An SRILM extension," in *Interspeech*, 2010.

[21] T. Alumäe and K. Kaljurand, "Maximum entropy language model adaptation for mobile speech input," in *Interspeech*, 2012.

[22] C. Chelba and A. Acero, "Adaptation of maximum entropy capitalizer: Little data can help a lot," *Computer Speech & Language*, vol. 20, no. 4, pp. 382–399, 2006.

[23] H. Xu, K. Li, Y. Wang, J. Wang, S. Kang, X. Chen, D. Povey, and S. Khudanpur, "Neural network language modeling with letter-based features and importance sampling," in *ICASSP*, 2018, pp. 6109–6113.

[24] Y. Wang, V. Peddinti, H. Xu, X. Zhang, D. Povey, and S. Khudanpur, "Backstitch: Counteracting finite-sample bias via negative steps," in *Interspeech*, 2017.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 2017-Decem, pp. 5999–6009, 2017.

[26] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006, pp. 369–376.

[27] J. H. Wong, Y. Gaur, R. Zhao, L. Lu, E. Sun, J. Li, and Y. Gong, "Combination of end-to-end and hybrid models for speech recognition," in *Interspeech*, 2020.

[28] A. Stolcke, Y. Konig, and M. Weintraub, "Explicit word error minimization in N-best list rescoring," in *Eurospeech*, 1997.