Interspeech 2025
17-21 August 2025, Rotterdam, The Netherlands

GigaAM: Efficient Self-Supervised Learner for Speech Recognition

Aleksandr Kutsakov, Alexandr Maximenko, Georgii Gospodinov, Pavel Bogomolov, Fyodor Minkin

SaluteDevices, Russia

{askutsakov, ae.maximenko, georgygospodinov, bobrosoft98, minkin.fyodor}@gmail.com

Abstract

Self-Supervised Learning (SSL) has demonstrated strong performance in speech processing, particularly in automatic speech recognition. In this paper, we explore an SSL pretraining framework that leverages masked language modeling with targets derived from a speech recognition model. We also present chunk-wise attention with dynamic chunk size sampling during pretraining to enable both full-context and streaming fine-tuning. Our experiments examine scaling with respect to model size and the amount of data. Using our method, we train the GigaAM family of models, including a state-of-the-art model for Russian speech recognition that outperforms Whisper-large-v3 by 50%. We have released our foundation and ASR models, along with the inference code, under the MIT license as open-source resources to the research community.

Index Terms: self-supervised learning, speech recognition, low-resource languages, foundation models, scaling laws

1. Introduction

Self-supervised learning (SSL) has transformed ASR by leveraging vast amounts of unlabeled data, reducing the need for manual annotation. Numerous methods have emerged [1, 2, 3, 4, 5, 6, 7], but some of them suffer from training instability due to representation collapse [1, 5, 7]. On the other hand, BEST-RQ [3] has proven effective due to its use of a frozen audio quantization module that provides targets for masked language prediction in a single stage. However, despite its training stability, BEST-RQ suffers from semantically uninformative targets because they are derived from low-level representations. Layer-wise analyses in the literature suggest that the most semantically rich features appear in the intermediate layers, while the final layers tend to exhibit an autoencoder-like behavior [8, 9].

Motivated by these observations, we propose an SSL pretraining framework that overcomes these limitations. Our approach leverages semantically enriched target variables derived from a supervised ASR model. Furthermore, to address potential model degeneracy on excessive silence in the audio data, we preprocess it by removing segments with insufficient speech activity. In contrast to PBERT [4], which extracts targets from an intermediate layer, we perform K-means clustering on the last layer of a CTC-based ASR model [10] to generate targets that capture higher-level semantic information.

In addition, we explored chunk size sampling during both pretraining and fine-tuning stages and found that a single pretraining run is sufficient to support both full-context and streaming ASR setups. Using our framework, we developed and publicly released the pretrained model with its fine-tuned variants for Russian ASR. Our models achieve state-of-the-art performance on Russian speech recognition across several benchmarks [11, 12, 13], outperforming competing models such as Whisper-large-v3[1] by nearly 50%, see Table 1.

Table 1: *Open-source models Word Error Rate (WER) on the Russian language datasets*

Model	Golos Farfield	Russian MCV-19	Russian LibriSpeech
XLS-R [14] finetuned	15.7	8.0	16.1
Whisper-large-v3	16.6	5.5	9.5
FastConformer-RNNT	6.6	5.7	11.3
Ours (CTC)	4.3	3.1	**5.5**
Ours (RNNT)	**3.9**	**2.7**	**5.5**

In summary, our key contributions are:

- **The development and public release[2] of a state-of-the-art Russian ASR model**, achieving substantial performance improvements over existing approaches.
- **A scalable self-supervised learning (SSL) framework** that utilizes semantically enriched targets generated by a CTC-based ASR model.
- **A dynamic chunking strategy** enabling unified pretraining for both full-context and streaming ASR within a single run.
- **A thorough scalability analysis** exploring the impact of labeled/unlabeled dataset sizes and model dimensions on performance.

2. Related Work

The field of self-supervised learning for speech encompasses a wide range of methods, from early approaches such as wav2vec [6] to more recent advances. Although many excellent techniques have been proposed, in this work we focus on two representative frameworks – HuBERT [1] and BEST-RQ – due to their complementary strengths. HuBERT, which forms the basis of our approach, employs a multi-stage masked language modeling pipeline to learn robust representations. At the same time, BEST-RQ is widely used in production-scale systems for its efficient, single-stage pretraining via a frozen audio quantizer [15]. In the following subsections, we detail these two approaches and discuss how our method builds upon and extends their core ideas.

[1] with forced Russian language token

[2] https://github.com/salute-developers/gigaam

10.21437/Interspeech.2025-1616

2.1. SSL Pretraining: HuBERT vs. BEST-RQ

In the HuBERT framework, discrete target tokens are generated through a multi-stage pipeline. Given an input sequence of audio features, a subset of time indices is selected for masking. At these masked positions, the input features are replaced with a predefined mask embedding. In Stage 1, K-means clustering is applied to low-level features (e.g., MFCCs) to assign each frame a token corresponding to its nearest centroid. In Stage 2, the clustering is repeated on the intermediate representations of a pretrained (teacher) model to yield refined assignments. By contrast, BEST-RQ employs random vector quantization with frozen codebooks and projection module. This approach avoids representation collapse issues observed in jointly trained models such as wav2vec2 [5] and w2v-BERT [16].

2.2. Importance of Data Domain in Pretraining

The pre-training data selection process is critical for downstream ASR performance. Prior work has shown that domain-specific models can outperform multilingual ones. For example, Babu et al. [14] demonstrated that monolingual models yield superior performance on domain-specific tasks, and Lu et al. [17] reported that filtering the pretraining data to match the target domain can lead to up to a 15% relative reduction in Word Error Rate (WER). Motivated by these findings, our approach pretrains exclusively on Russian speech to ensure that the learned representations are optimally adapted to the target domain.

2.3. Chunkwise Attention Mechanisms

In practice, ASR models are often trained on relatively short audio segments (e.g., up to 30 seconds), which can be problematic when operating in a long-form scenario. For instance, a full-context model trained on 30-second audio fragments might struggle when presented with a 2-minute audio file. To address this, we employ a chunkwise attention mechanism that divides the input into fixed-length segments.

Formally, given an input sequence and a predefined chunk size (e.g., 200ms, 1s, 2s, 4s, or 8s), we partition the sequence into consecutive segments of the selected length. Within each chunk, attention is computed locally, enabling the model to focus on limited context rather than the entire sequence. This not only helps in generalizing to longer audio inputs but also supports streaming scenarios where only limited future context is available. We further incorporate dynamic chunk size sampling during pretraining so that the model learns to adapt to varying context lengths, allowing it to perform effectively in both full-context and streaming setups without requiring expensive re-training runs.

2.4. Scaling Laws in SSL Pretraining

Scaling laws have been extensively studied in natural language processing [18, 19] and computer vision [20], yet systematic investigations for self-supervised ASR remain limited. While previous works [21, 22, 23] has examined the influence of model capacity, the amount of training data, and compute budget in multilingual and semi-supervised settings, a comprehensive analysis for self-supervised pretraining is still needed. In this work, we analyze scaling effects by varying the number of pretraining steps, the pretraining dataset size (spanning two orders of magnitude), the fine-tuning dataset size (spanning three orders of magnitude), and the model capacity (spanning 1.5 orders of magnitude), and we observe how these factors influence the performance of our SSL model on downstream task.

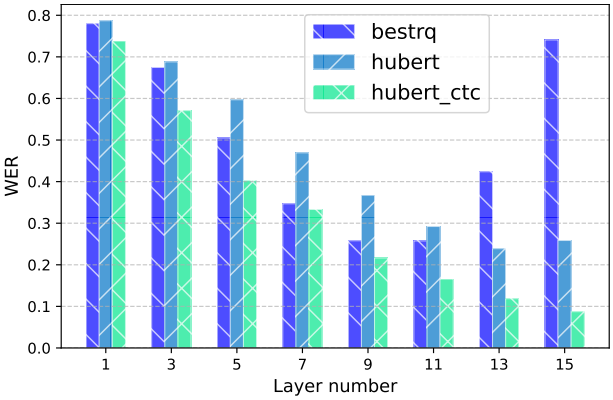

Figure 1: *Intermediate layers probing of pretrained encoders with different pretraining approaches*

3. Our methodology

3.1. Pretraining method

Our pretraining method HuBERT-CTC builds on HuBERT, where KMeans clustering is applied to the hidden states of the encoder. However, unlike the standard HuBERT approach, which uses a pretrained encoder, we employ an encoder fine-tuned on ASR tasks. We compare the properties of these approaches by probing the intermediate layers of the pretrained encoders on speech recognition. As shown in Fig. 1, the BEST-RQ/HuBERT pretrained encoders achieve the best performance at the intermediate layers, with quality degrading towards the last layers. In contrast, our method demonstrates a monotonic improvement in ASR performance up to the latest layer. This results in more ASR-specific representations being learned, which is particularly crucial in low-resource settings.

3.2. Data preprocessing

3.2.1. Voice Activity Detection filtering

To improve the quality of our pretraining data, we employ Voice Activity Detection (VAD) filtering to identify and remove segments with excessive silence, similar to [24, 15]. Each audio sample from the pretraining dataset was implicitly split into one-minute chunks to avoid discarding large samples during filtering. The segments in which the proportion of silence exceeded 60% were discarded. This process retained 80% of the original data while improving the model performance by an average of $5 - 10\%$.

3.2.2. More advanced data filtering methods

In addition to VAD filtering, we explore more sophisticated data filtering techniques, including methods from SemDeDup [25] and data pruning [26]. These algorithms were originally designed for language models and aim to reduce redundancy and improve the diversity of training data. We applied them both separately and on top of VAD-filtered data, however, the observed changes in model performance were negligible. Given the significant computational overhead of these methods and their limited impact, we decided to rely solely on VAD filtering for our data preprocessing pipeline.

3.3. Streaming and long-form inference

Chunkwise attention enables the model to limit its context size and process audio of varying lengths, different from those seen during training. By using a sufficiently large chunk size of several seconds, the receptive field can extend to approximately a minute. Streaming applications are achieved by reducing the chunk size and employing chunkwise causal convolutions [27] in the conformer layers, ensuring that the required future context is limited to the chunk size and allowing the reuse of pretrained non-streaming version.

4. Experiments

Our experimental setup is as follows: for self-supervised pretraining with HuBERT-CTC, we use 100k hours of Russian audio and train for 400k steps with a virtual batch size of 9 hours. For supervised fine-tuning, we utilize datasets including Golos [11], Mozilla Common Voice (ru) [12], Russian LibriSpeech [13], and SOVA [28], totaling 2k hours of audio, and train for 100k steps with a batch size of 2048 samples. Across all experiments, we employ the Conformer [29] architecture with a standard model size of 240M parameters. The 240M teacher model for HuBERT is trained on 50k hours of pretraining audio using default hyperparameters and the wav2vec2 approach. HuBERT-CTC utilizes a fine-tuned version of this model with the same supervised data as the other models in the paper. To compute the final WER in the following experiments, we average the metrics over the last ten epochs on the test sets (Golos Farfield/Crowd, MCV, LibriSpeech). For few-shot supervised data, where overfitting is likely, we employ early stopping.

4.1. Other pretraining methods

To compare our approach with other pretraining methods, including the setup without any pretraining at all, we finetune the model with different amounts of supervised data (100%, 10%, 1%, 0.1%). Our results are presented in Table 2. Even when using the full dataset, we observe a significant performance gap between our method and others, which further widens as the amount of supervised data decreases. This demonstrates the ability of HuBERT-CTC to distill knowledge from the ASR model even on unlabeled data, highlighting its applicability in low-resource scenarios.

Table 2: *The final WER for different pretraining methods and fine-tuning set sizes*

	Fine-tuning set size			
Pretraining	**100%**	**10%**	**1%**	**0.1%**
w/o pretraining	5.71	21.12	76.78	–
BEST-RQ	5.18	6.47	9.90	16.80
Wav2vec2	4.62	6.6	10.66	19.38
HuBERT	3.66	5.19	8.48	16.70
HuBERT-CTC	3.35	4.44	5.77	7.41

4.2. Data, model and steps scaling

We conduct a comprehensive investigation into the relationship between model performance and the amount of (un)labeled data, model capacity, and computational budget. To provide clarity, we outline all configurations used in our experiments:

- Pretraining dataset size: 1k, 6k, 12k, 24k, 50k, 100k hours.

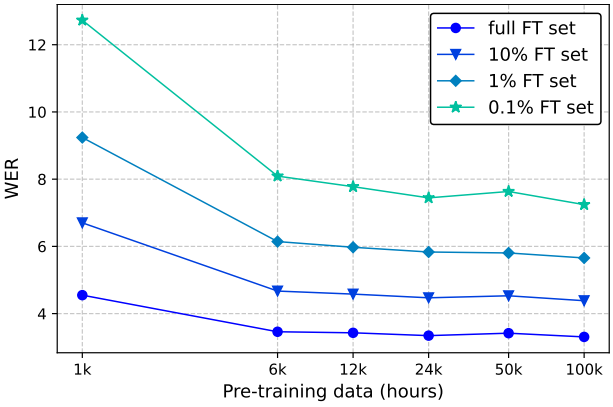

Figure 2: *The dependency of WER on the pretraining dataset size with fine-tuning set scaling*

- Fine-tuning dataset size: $2000, 200, 20, 2$ hours.
- Model size: 30M, 60M, 100M, 240M, 500M.
- Number of pretraining steps: $25k, 100k, 400k$ steps.

4.2.1. Pretraining data

The reduction of supervised data quantity significantly affects speech recognition quality, but the dependence on the amount of unsupervised data is less frequently explored. The results of our investigation, spanning from 1k to 100k hours of unsupervised audio, are presented in Figure 2. A key property of the HuBERT-CTC approach is the stabilization of model performance after reaching a certain threshold of pretraining data. Specifically, for values above 6k hours, the final quality remains nearly unchanged, as the student model effectively transfers knowledge from the teacher even with a relatively small dataset. This property will be particularly advantageous for distilling large multilingual models into smaller, domain-specific ones, and we leave further exploration of this direction as future work. Notably, a significant degradation in quality occurs only when the unsupervised data is reduced to just 1k hours, which is half the size of the supervised dataset.

4.2.2. Model size

We investigate the relationship between model performance and size, exploring models ranging from 30M to 500M parameters. The amount of supervised data is varied similarly to the previous experiment. The results are presented in Figure 3. Since varying the amount of pretraining data from 6k to 100k hours yields only minor changes in performance, we compute the mean across these five configurations and include the standard deviation on the graph. Remarkably, a 100M model outperforms the 240M teacher model ($4.62 \rightarrow 4.21$ WER), even when pretrained on a smaller amount of data.

4.2.3. Pretraining as regularization

We also explored fine-tuning regularization techniques aimed at accelerating convergence and improving model generalization. Among these, the CR-CTC [30] method showed particular promise, significantly enhancing the final performance of smaller models in the absence of pretraining. Table 3 presents the results in terms of relative WER improvement (in percentage) after applying CR-CTC for different model sizes and pre-

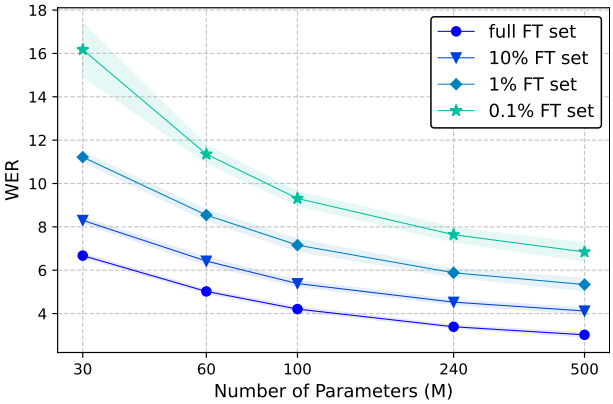

Figure 3: *Quality dependency on model size with fine-tuning set scaling. For each configuration, we average the metrics across pre-training data quantities ranging from* $6k$ *to* $100k$ *hours and include the standard deviation on the graph*

training durations. Notably, the application of CR-CTC does not benefit larger models, while for smaller models, pretraining effectively replaces the need for regularization, resulting in only marginal changes when CR-CTC is applied.

Table 3: *CR-CTC impact comparison with partial pretraining, given as a percentage improvement w/o* $\rightarrow$ *w/ CR-CTC. The mean and standard deviation are calculated across pretraining data quantities ranging from* $6k$ *to* $100k$ *hours. The column labeled "–" corresponds to training from scratch without pretraining*

Model	**Pretraining steps**			
	–	**25k**	**100k**	**400k**
30M	17.0	15.7 ± 0.7	3.0 ± 0.5	1.5 ± 0.4
60M	14.2	13.4 ± 3.8	2.0 ± 1.0	-0.7 ± 1.3
100M	13.5	9.6 ± 1.8	3.3 ± 1.8	-0.6 ± 0.5
240M	4.4	-3.3 ± 1.4	-0.4 ± 1.8	-3.1 ± 0.4
500M	-6.7	-3.3 ± 1.1	0.1 ± 0.8	0.6 ± 0.9

4.3. Streaming and long-form inference

In our experiments we utilize chunkwise attention with chunk sizes of 200ms, 1s, 2s, 4s, and 8s, as well as dynamic chunking, where chunk sizes were uniformly sampled from these values. During pretraining, a chunk size of 200ms is excluded, as the training process tends to diverge. Table 4 presents the results of various pretraining chunking configurations fine-tuned with specific chunk sizes, including both chunkwise causal and standard convolutions. For the 200ms chunk, only streaming convolutions are used, as this size is impractical for long-form. The dynamic chunking proves to be the most versatile approach for pretraining, consistently delivering some of the best results even when fine-tuning (and during inference) with full-context attention. We also experimented with dynamic chunk size sampling during fine-tuning, but this led to a degradation in quality.

4.3.1. Limited kernel size

To limit the receptive field increase to one chunk per layer, we employ a small kernel size of 5 (200ms) for convolutions in our model. We conducted an ablation study with a larger kernel size of 31 ($>$ 1s) and observed a slight degradation in performance ($3.35 \rightarrow 3.49$ WER) compared to the smaller kernel size. This result allows us to use a small kernel size without losing quality or unnecessarily expanding the context size.

Table 4: *Chunking results: the "pretraining" column specifies the pretraining chunking method used. For convenience, results are grouped by the fine-tuning chunk size (full-context option is given by "–"). "Long-form" refers to standard convolutions, "streaming" corresponds to chunkwise causal convolutions, where the future context is restricted to one chunk*

Pretraining	**FT chunk**	**WER**	
		Long-form	**Streaming**
Full-context	–	3.35	–
Dynamic chunk	–	3.31	–
Full-context	8s	3.47	3.52
Chunk=8s		3.39	3.39
Dynamic chunk		3.39	3.44
Full-context	4s	3.92	4.02
Chunk=8s		3.56	3.83
Chunk=4s		3.58	3.85
Dynamic chunk		3.51	3.77
Full-context	2s	4.42	5.03
Chunk=8s		3.91	4.66
Chunk=4s		3.83	4.7
Chunk=2s		3.62	4.49
Dynamic chunk		3.63	4.56
Full-context	1s	5.09	6.37
Chunk=8s		4.32	5.88
Chunk=4s		4.2	5.81
Chunk=2s		3.94	5.56
Chunk=1s		3.94	5.68
Dynamic chunk		3.97	5.49
Full-context	200ms	–	12.3
Chunk=8s		–	11.71
Chunk=4s		–	10.98
Chunk=2s		–	10.65
Chunk=1s		–	10.35
Dynamic chunk		–	10.37

5. Conclusion

We presented an SSL pretraining framework for ASR which leverages semantically enriched targets from a CTC-based ASR model. By integrating dynamic chunk size training, our method supports both full-context and streaming fine-tuning ability within a single pretraining run. Experiments on Russian speech recognition demonstrate that our HuBERT-CTC method outperforms BEST-RQ, HuBERT, and wav2vec2 – with our proposed models achieving state-of-the-art performance and surpassing Whisper-large-v3 by nearly 50%. It is also worth noting that all other presented models utilize autoregressive decoding, and many of them are significantly larger than ours. Despite this, they are outperformed by even the non-autoregressive CTC version of our model. Our scaling analysis confirms the robustness of the approach.

6. References

[1] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[2] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, M. Zeng, and F. Wei, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, pp. 1505–1518, 2021.

[3] C. Chiu, J. Qin, Y. Zhang, J. Yu, and Y. Wu, "Self-supervised learning with random-projection quantizer for speech recognition," in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 2022, pp. 3915–3924.

[4] C. Wang, Y. Wang, Y. Wu, S. Chen, J. Li, S. Liu, and F. Wei, "Supervision-guided codebooks for masked prediction in speech pre-training," in *Interspeech 2022*, 2022, pp. 2643–2647.

[5] 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*, vol. 33. Curran Associates, Inc., 2020, pp. 12 449–12 460.

[6] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proc. of Interspeech*, 2019.

[7] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "data2vec: A general framework for self-supervised learning in speech, vision and language," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 17–23 Jul 2022, pp. 1298–1312.

[8] A. Pasad, J.-C. Chou, and K. Livescu, "Layer-wise analysis of a self-supervised speech representation model," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 914–921.

[9] Z. Huo, K. C. Sim, B. Li, D. Hwang, T. N. Sainath, and T. Strohman, "Resource-efficient transfer learning from speech foundation model using hierarchical feature fusion," *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2022.

[10] A. Graves, S. Fernández, F. J. 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*, ser. ACM International Conference Proceeding Series, vol. 148. ACM, 2006, pp. 369–376.

[11] N. Karpov, A. Denisenko, and F. Minkin, "Golos: Russian dataset for speech research," in *Interspeech 2021*, 2021, pp. 1419–1423.

[12] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 4218–4222.

[13] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "Mls: A large-scale multilingual dataset for speech research," in *Interspeech 2020*, 2020, pp. 2757–2761.

[14] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli, "Xls-r: Self-supervised cross-lingual speech representation learning at scale," in *Interspeech 2022*, 2022, pp. 2278–2282.

[15] Y. Zhang, W. Han, J. Qin, Y. Wang, A. Bapna, Z. Chen, N. Chen, B. Li, V. Axelrod, G. Wang, Z. Meng, K. Hu, A. Rosenberg, R. Prabhavalkar, D. S. Park, P. Haghani, J. Riesa, G. Perng, H. Soltau, T. Strohman, B. Ramabhadran, T. Sainath, P. Moreno, C.-C. Chiu, J. Schalkwyk, F. Beaufays, and Y. Wu, "Google usm: Scaling automatic speech recognition beyond 100 languages," 2023. [Online]. Available: https://arxiv.org/abs/2303.01037

[16] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, and Y. Wu, "w2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 244–250.

[17] Z. Lu, Y. Wang, Y. Zhang, W. Han, Z. Chen, and P. Haghani, "Unsupervised data selection via discrete speech representation for asr," in *Interspeech 2022*, 2022, pp. 3393–3397.

[18] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling Laws for Neural Language Models," 1 2020.

[19] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, O. Vinyals, J. W. Rae, and L. Sifre, "Training compute-optimal large language models," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22. Red Hook, NY, USA: Curran Associates Inc., 2022.

[20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[21] B. Li, R. Pang, T. N. Sainath, A. Gulati, Y. Zhang, J. Qin, P. Haghani, W. R. Huang, M. Ma, and J. Bai, "Scaling end-to-end models for large-scale multilingual asr," 2021.

[22] W. Zheng, A. Xiao, G. Keren, D. Le, F. Zhang, C. Fuegen, O. Kalinli, Y. Saraf, and A. Mohamed, "Scaling asr improves zero and few shot learning," in *Interspeech 2022*, 2022, pp. 5135–5139.

[23] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 23–29 Jul 2023, pp. 28 492–28 518.

[24] K. Zhang, L. Chkhetiani, F. M. Ramirez, Y. Khare, A. Vanzo, M. Liang, S. R. Martin, G. Oexle, R. Bousbib, T. Peyash, M. Nguyen, D. Pulliam, and D. Donato, "Conformer-1: Robust asr via large-scale semisupervised bootstrapping," 2024.

[25] R. Ardila, M. Branson, K. Davis, M. Henretty, and M. K. et al, "Semdedup: Data-efficient learning at web-scale through semantic deduplication," *Submitted to Transactions on Machine Learning Research*, 2023, rejected.

[26] B. Sorscher, R. Geirhos, S. Shekhar, S. Ganguli, and A. Morcos, "Beyond neural scaling laws: beating power law scaling via data pruning," in *Advances in Neural Information Processing Systems*, vol. 35. Curran Associates, Inc., 2022, pp. 19 523–19 536.

[27] X. Li, G. Huybrechts, S. Ronanki, J. Farris, and S. Bodapati, "Dynamic chunk convolution for unified streaming and non-streaming conformer asr," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[28] E. Zubarev and T. Moskalets, "Sova dataset," https://github.com/sovaai/sova-dataset, 2022.

[29] 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*, 2020, pp. 5036–5040.

[30] Z. Yao, W. Kang, X. Yang, F. Kuang, L. Guo, H. Zhu, Z. Jin, Z. Li, L. Lin, and D. Povey, "Cr-ctc: Consistency regularization on ctc for improved speech recognition," 2024.