INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

An efficient approach for the automated segmentation and transcription of the People's Speech corpus

Astik Biswas[1], *Abdelmoumene Boumadane*[2], *Stephane Peillon* [2], *Gildas Bleas* [2]

[1]Oracle Cloud Infrastructure, India
[2]Oracle Cloud Infrastructure, France

astik.biswas@oracle.com, abdelmoumene.boumadane@oracle.com, stephane.peillon@oracle.com, gildas.bleas@oracle.com

Abstract

Advancements in speech technology have led to the integration of modern ASR systems into various applications such as chatbots, medical dictation, video transcription etc. Conversational ASR training requires speech that captures the acoustic cues of spontaneous speech. With its 30k hours of conversational speech, the People's Speech corpus is the largest available spontaneous and conversational corpus and an invaluable resource for such training. In addition, it comes with a commercial friendly license. The corpus is packaged in uniform 15-second segments, but this can lead to abrupt cutting off of speech and transcription that is not always accurate. This paper presents an effective method for automatic data mining from a small subset of 973 raw original records used by the People's Speech corpus. The paper also proposes an approach for outlier detection and automatic data curation. Results show a 19.7% relative improvement in WER compared to the original segments.

Index Terms: ASR, wav2vec2, People's Speech, Data curation

1. Introduction

In recent years, advancements in deep learning technology have enabled the development of highly accurate Automatic Speech Recognition (ASR) systems that can recognize speech across various domains, accents, speakers, and environments. However, the community is working to improve ASR technology even further for efficient conversational AI. To develop an efficient ASR model suitable for conversational AI, we need high-quality training data that comprises conversational and spontaneous speech from diverse environments and a wide variety of speakers. However, most of the large corpora available today, such as MLS [1], Librispeech [2], and CommonVoice [3], consist of read speech that lacks spontaneous speech properties. To train a good conversational ASR model, it is important to use spontaneous speech that includes fast speech and hesitations. While Whisper [4] has shown the potential of using 600K hours of semi-supervised data for ASR, it is not feasible for startups and academia to use such a large volume of data. Therefore, it is important to focus on the quality of the training data rather than the volume.

To the best of our knowledge, the Gigaspeech [5] and People's Speech [6] corpora are the only large-scale conversational and diverse speech corpora that contain 10K and 30K hours of speech, respectively. These two corpora are valuable training resources, but People's Speech alone is available both to commercial applications and to the research community. However, People's Speech transcription is not always accurate. First, it is packaged in uniform 15-second segments, which often abruptly truncate speech at the boundaries, leading erroneous transcription at those points. This is also not ideal to train an ASR on fixed chunked width, as this may have an adverse effect on recognizing short or single-word audio, especially for end-to-end ASR. Second, the transcriptions provided with the corpus are not always reliable and were often generated using pre-trained ASR models. We've also noticed that many segments contain non-English speech or music, resulting in incorrect alignments. Therefore, there is a lot of room for improvement in terms of transcript quality, including deploying new state-of-the-art ASR models and exploring resegmentation techniques to create corpora with variable segment lengths.

In this paper, we present an alternative solution to the time-consuming and expensive process of curating large-scale data by human experts. Instead, we propose to discard the People's Speech transcriptions and to recreate the data from the original *archive.org* records, which were used to create the distributed corpus. Our method involves a pipeline that automatically re-segments and retranscribes the raw audio data. The process includes transcription, punctuation, segmentation from the said punctuation, frame-level alignment of segments, retranscription of updated segments, final alignment from the latest transcription, and final resegmentation based on silences. We demonstrated the effectiveness of this process on a subset of 973 raw audio archives (of 55 minute length in average), sourced from archive.org. Although our pipeline employs in-house models (for better control and accuracy in downstream tasks such as word-level timestamp calculation), the same process could be deployed only with open-source models, and used on a large scale. To the best of our knowledge, this is the first attempt to recreate this data set with improved quality using a standard CPU machine.

In addition, we propose a solution to data curation by leveraging automatic outlier detection and curation based on the wav2vec2 model [7, 8]. First, we use an in-house model to calculate the alignment loss [9] of the given transcription and thereby detect any possible outliers. Then we introduce a third-party pre-trained model (curator) to further curate those outliers based on their connectionist temporal classification (CTC) [10] loss value. Our strategy demonstrates the potentiality of automatic segmentation and curation in improving the quality of large-scale speech data.

2. People's Speech corpus

The People's Speech dataset is a public dataset that includes diverse sources of speech beyond audiobooks, such as movies, TV, local news, sports commentary, music, historical documentaries, video game replays, and many more domains, along with said ground-truth transcripts from subtitles. However, these transcripts may not always be reliable, as they may not be

10.21437/Interspeech.2023-1208

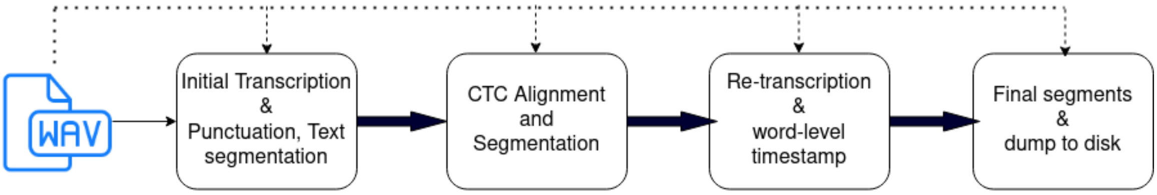

WAV
Initial Transcription & Punctuation, Text segmentation
CTC Alignment and Segmentation
Re-transcription & word-level timestamp
Final segments & dump to disk

Figure 1: *People's Speech re-segmentation and re-transcription pipeline*

created by human experts and may contain translations, summaries, or notes instead of the actual speech in the audio. The corpus creation process involves automatic transcription with word-level timestamps, segmentation into 15-second uniform segments, forced alignment, and verification. Based on validation assumptions that include analyzing character error rate (CER) and word error rate (WER) with respect to ground truth transcriptions, the entire People's Speech dataset is subcategorized by authors [6] into three parts: 10K hours of gold quality, 7K hours of high quality, and 14K hours of low quality. However, we believe that using WER/CER as a measure of quality is not always reliable unless the text is perfectly normalized and the ground truth is accurate.

In the following sections, we will present our solution for producing better quality segments of varying lengths by utilizing audio files from *archive.org*, which were used in the creation of the People's Speech corpus.

3. Segmentation and Transcription Pipeline

The pipeline involves both in-house and third-party models. We used in-house models for the initial transcription and final alignments, while we used third-party models from Nemo [11] for the remaining tasks. However, the in-house models can be substituted with third-party models, such as Whisper or Nemo models that are well suited for the task. We opted for third-party models for the final transcription to prevent any bias in future model training and testing on this new data. The process is illustrated in Figure 1.

3.1. Initial long audio transcription

To begin with, we employed an in-house model to obtain the initial transcription for a long audio file. We decided to use automatic transcription as provided transcription is not enough reliable. After transcription, we used a pre-existing tool to restore punctuation[1] and divided the transcript into sentences or speech sections. These sections were used as input for the next step (CTC alignment). This was to have better cut between words, ideally during silences that appear after a comma or period, but also to ease the alignment on short segments. As the initial transcription generated by the ASR system had already been normalized i.e spoken form, text normalization process was not required.

3.2. CTC forced alignment and Segmentation

This stage of the pipeline utilizes CTC-segmentation [9], an algorithm that extracts appropriate audio-text alignments in the presence of unknown speech sections at the beginning or end of the audio recording. The algorithm utilizes a CTC-based end-to-end network that is trained on pre-aligned data. For this task, we used the Nemo citrinet model[2] [12], which is faster and uses less memory compared to other models such as conformer-based models based on our experience. Based on the alignment score (-10), we selected the audio segments that were passed on to the next stage in the pipeline for more accurate re-transcription. The duration of the audio segments ranged from a few seconds to one minute.

3.3. Re-transcription and word-level alignment

The final transcription of each speech section was obtained by using the Nemo conformer-CTC model[3], followed by calculating word-level timestamps. The choice of conformer-CTC was based on its reported higher accuracy compared to Citrinet model, as well as our observation of less deletions compared to the former. The conformer model also has reasonable speed and memory consumption for short audio segments. The output of this stage was the final transcription and its corresponding word-level timestamps for each speech section.

3.4. Creation of final segments

In the final step of the pipeline, several rules were followed to create "better final segments", splitting too long ones using the time-stamp information and utilizing available silences at the boundaries. These rules include:

1. Creating a list of segments from the word level timestamp. The minimum pause of 200 ms was allowed between each segment.
2. Splitting segments to conform to a maximum number of characters per segment (200) and the maximum allowed duration (15 seconds).
3. Looking for silences between two consecutive segments. If there is enough silence (more than 1000 ms), adjusting the segment start and end boundaries to keep only 800 ms of silence at the start and/or end.

The final step was to dump segmented WAV files along with their corresponding transcriptions. At last, we ended of with 543 hours of data (204K records) obtained from 973 raw audio files, and entire process took us approximately three days to complete on machine equipped with an Intel Xeon processor containing 104 CPUs @ 2.00GHz. The process did not involve a GPU, and we can anticipate a significant speed boost by utilizing a GPU.

4. Automatic outlier detection and curation

The forced alignment method discussed in section 3.2 can generate precise alignments between provided transcripts and audio snippets. However, spontaneous speech, which includes slips of the tongue and stutters, can complicate the transcription and

[1] `https://github.com/oliverguhr/deepmultilingualpunctuation`

[2] `https://huggingface.co/nvidia/stt_en_citrinet_1024_gamma_0_25`

[3] `https://huggingface.co/nvidia/stt_en_conformer_ctc_large`

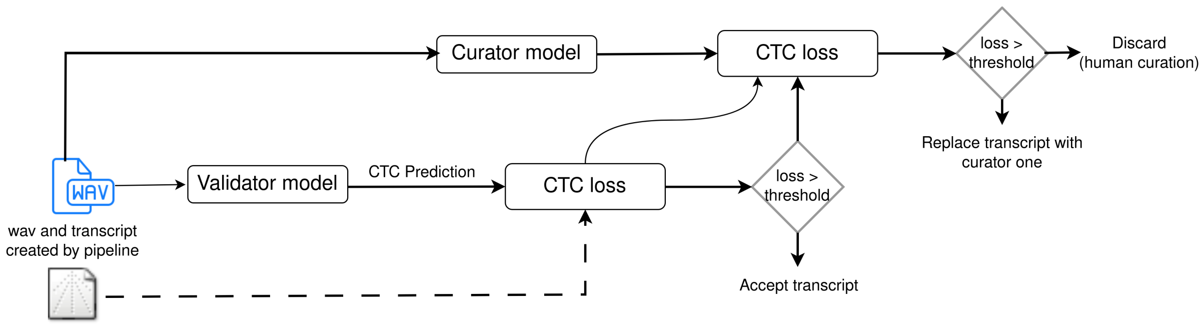


Figure 2: *Automatic curation pipeline*

alignment process. Any ASR systems used for re-transcription are error-prone, and depending on the quality of data on which they were trained, or the normalization applied on these data, they may make recurring errors that will then be learned by our final model. As a result, downstream errors can occur when the transcript differs from what is actually said, making the data noisy and unsuitable for acoustic training. Detecting outliers can help improve data quality. In this section, we demonstrate how we use CTC alignment loss to identify possible outliers and curate them using another third-party model, based on a loss threshold. The process is illustrated in Figure 2.

We implemented a validator model to compute CTC prediction matrix for the newly created segments, and we calculated the CTC loss with respect to the new transcription. Segments with high loss values were considered outliers, as they differed considerably from the validator model predictions. We analyzed around 200K newly created segments, and the CTC loss values are summarized in Table 1. We observed that the level of disagreement between the reference transcription and validator could go high when the CTC loss was high, and it could exceed 1000. We hypothesized that this could be due to a difficult segment that the model couldn't predict well or an erroneous transcription. In the case of an erroneous transcription, we attempted to curate it using another model to minimize the CTC loss. We used Nemo RNN-T model[4] as a curator and applied a loss threshold of 50. We were able to curate 70% of the records using this strategy.

Table 1: *CTC loss calculated by validator*

mean	std	min	25%	50%	75%	max
26.22209	34.37976	0.00018	4.35161	13.81459	34.54270	1075.92834

5. Experiments

5.1. Data

We trained six models using various combinations of data, as shown in Table 2. We used three different versions of People's Speech data, each sourced from the same original audio. These versions are referred to as:

- **v1**: A subset of the original People's Speech data, utilized in *Data1* and *Data1′*.
- **v2**: The re-segmented and re-transcribed from archive.org, used in both *Data2* and *Data2′*. Used same audio source from *archive.org* which used for **v1** in the People's Speech.
- **v3**: An automatically curated version of **v2**, used in *Data3* and *Data3′*.

The table illustrates that we were able to extract approximately 543 hours of training data for People's Speech by using 973 records. By applying automatic curation, we were able to discard five hours of training data from the re-transcribed training pool (data2). To make the training set more diverse, we included CommonVoice data in our training pool. This was done to observe the impact of newly segmented and transcribed data when combined with data from other sources or domains.

We used several publicly available datasets, including version 9 of Common Voice [3], financial domain data Earnings-21 [13], medical data Primock57 [14], and LibriSpeech [2], to evaluate our models.

Table 2: *Data composition used in acoustic modelling*

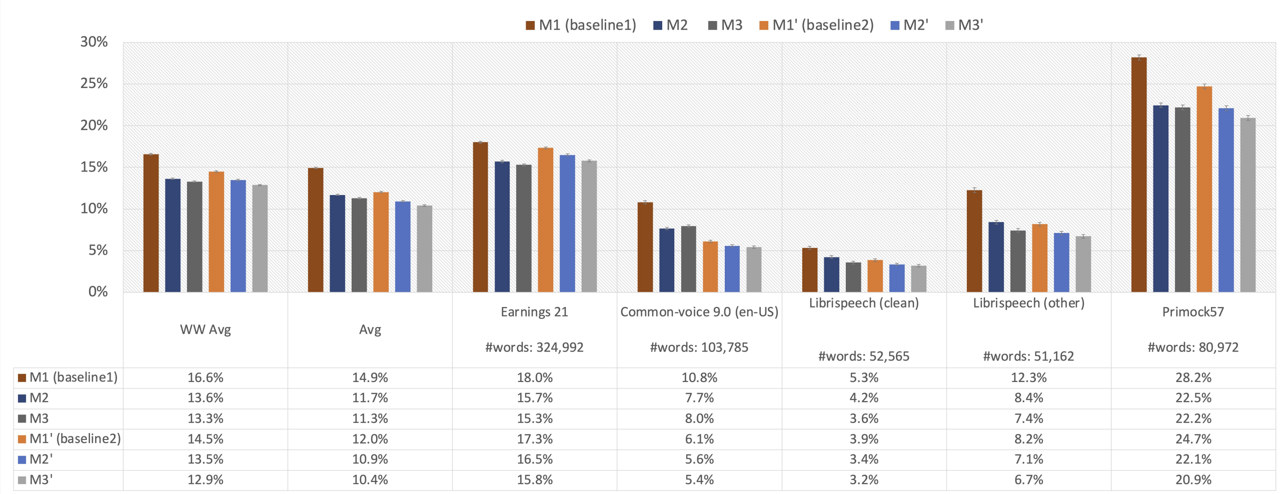
Training	Archive (h)		CommonVoice	Total
set	Resegmented	People'sSpeech	(h)	(h)
Data1	-	538.4 (v1)	-	538.4
Data2	543.0 (v2)	-	-	543.0
Data3	538.9 (v3)	-	-	538.9
Data1′	-	538.4 (v1)	623.7	1162.1
Data2′	543.0 (v2)	-	623.7	1166.7
Data3′	538.9 (v3)	-	623.7	1162.6

5.2. Acoustic Modelling

In our experiments, we utilized the wav2vec2 model [7] to transform raw audios into higher-level contextual features or embeddings via a set of convolutional layers that capture local features. Then, a proportion of these local features are masked and sent to a contextual transformer network for predicting the masked features using contextual information. The model was trained with a contrastive objective, similar to the BERT [15] model, which measures the model's ability to differentiate between a true masked input segment and a set of distractors. The self-attention layers in the transformer help encode information from the context surrounding a given masked segment. We used the publicly available pre-trained wav2vec2[5], which was trained on multiple data sources, including Libri-Light, CommonVoice, Switchboard, and Fisher. This model, consisting of 53 languages and 56K hours of speech data, was fine-tuned on the data described in Table 2 by adding three randomly initialized linear layers to the pre-trained model and then training with character-level CTC loss [10]. We used a learning rate (LR) of 1e-5 for the wav2vec2 model and a LR of one for the linear layers. We used Speechbrain toolkit [16] to train the models for 16 epochs with a batch size of 32, and we used the standard **27** English characters, including blank. A machine that is equipped with eight A100 GPUs took an average of 3.5 days to complete each model. After fine-tuning the model, we decoded on different test sets to calculate the corresponding WER. We trained six different models, each with a size of 315 million

[4] `https://huggingface.co/nvidia/stt_en_conformer_transducer_xlarge`

[5] `https://huggingface.co/facebook/wav2vec2-large-robust`



	WW Avg	Avg	Earnings 21 #words: 324,992	Common-voice 9.0 (en-US) #words: 103,785	Librispeech (clean) #words: 52,565	Librispeech (other) #words: 51,162	Primock57 #words: 80,972
■ M1 (baseline1)	16.6%	14.9%	18.0%	10.8%	5.3%	12.3%	28.2%
■ M2	13.6%	11.7%	15.7%	7.7%	4.2%	8.4%	22.5%
■ M3	13.3%	11.3%	15.3%	8.0%	3.6%	7.4%	22.2%
■ M1' (baseline2)	14.5%	12.0%	17.3%	6.1%	3.9%	8.2%	24.7%
■ M2'	13.5%	10.9%	16.5%	5.6%	3.4%	7.1%	22.1%
■ M3'	12.9%	10.4%	15.8%	5.4%	3.2%	6.7%	20.9%

Figure 3: *Performance on each test set using different models. The total number of word in each test set is reported in first row*

parameters. The first three models, namely **M1** (using data1), **M2** (using data2), and **M3** (using data3), were trained without including CommonVoice in the training data. The other three models, **M1**$'$ (using data1$'$), **M2**$'$ (using data2$'$), and **M3**$'$ (using data3$'$), were trained with CommonVoice included in the training pool. The models **M1** and **M1**$'$ can be considered as baselines because the People's Speech data used for training these models comes from the original distribution.

6. Results & Discussions

ASR performance was evaluated measuring WER on each of the test set mentioned in section 5.1. The results with the different models are reported in figure 3. The average across WER is not always reliable to understand the performance as it may be biased due to the size of each test set. Thus, we report word weighted average WER (WW Avg) (1^{st} column) and average WER (Avg) (2^{nd} column).

When comparing M1 and M2, improvements can be seen across all test sets, including conversational datasets such as Earnings and Primock57. Specifically, M2 demonstrates a relative improvement of 12.8% and 20.8% for Earnings and Primock, respectively, compared to M1. Overall, M2 shows a 17% relative improvement in WW Avg compared to M1, indicating the positive impact of re-segmented People's Speech and new transcriptions on the acoustic model. The significant improvement with the medical test set, i.e Primock, highlights the potential use of good conversational speech data like People's Speech. Effective segmentation allows the model to better learn using different segment lengths compared to the original uniform People's Speech segments. Similar trends can be observed when comparing the performance of models M1$'$ and M2$'$, with improvements observed in read and conversation speech data, including primock data. Specifically, M2$'$ shows a relative improvement of 5% and 10.5% for Earnings and Primock57, respectively, compared to M1$'$. Overall, a relative improvement of 6.9% in WW Avg is achieved with M2$'$ over M1$'$.

Focusing on the automatic outlier detection and curation process described in section 4, we can observe that it helps to reduce the word error rate (WER) even further. When comparing models M2 and M3, this process significantly minimize the WER across the test set, except for Commonvoice. For the conversational test sets, Earnings and Primock57, there is a relative improvement of 2.5% and 1.2% with M3 compared to M2, and the overall improvement in the word-weighted average (WW Avg) is 2.4% relative. Similarly, model M3$'$ also follows this trend and offers an overall relative improvement of 4.5% in WW Avg compared to M2$'$, and this improvement can be observed across all the test sets, including Primock57 and Earnings. When comparing M3 and M3$'$ with their respective baselines, M1 and M1$'$, the overall relative WW Avg improvements are significant at 19.7% (3.2% absolute) and 11.1% (1.6% absolute), respectively. These results highlight the benefits of leveraging a good resource of conversational speech corpus, such as People's Speech, with effective automatic re-segmentation and curation for acoustic model training.

7. Conclusion

We outlined a solution for efficiently re-segmenting and transcribing conversational speech data, specifically the People's Speech dataset. The process involves multiple stages, such as transcription, punctuation restoration and text segmentation forced alignment to determine segment-level boundaries, retranscription of these segments with word-level time-stamp calculation, and generating final audio files based on a set of fixed rules. We used both in-house and open-source models to complete this pipeline, but noted that the in-house models could be substituted with open-source alternatives to achieve the same outcome. The study utilized 973 raw audio files from the *archive.org*, resulting in approximately 543 hours of data. We demonstrated the effectiveness of this process by testing different acoustic models that highlights its potential to boost up ASR performance across domains. The pipeline can be adapted to extract speech data from any raw audio and used for speech recognition tasks across languages. Furthermore, the authors implemented an automatic outlier detection process based on CTC loss to improve the quality of silver or bronze level dataset. In the future, we plan to optimize the re-segmentation pipeline and expand the mining process to create a larger-scale corpus from the People's Speech raw audio archive.

8. Acknowledgement

We are thankful to Phanindra, Serge and Nancy for their input and suggestion to complete the paper. We are also thankful to Oracle cloud for providing the infrastructure for this study.

9. References

[1] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "MLS: A Large-Scale Multilingual Dataset for Speech Research," in *Proc. Interspeech 2020*, 2020, pp. 2757–2761.

[2] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*. IEEE, 2015, pp. 5206–5210.

[3] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in *Proc. LREC*, 2019, pp. 4218–4222.

[4] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022.

[5] G. Chen, S. Chai, G. Wang, J. Du, W.-Q. Zhang, C. Weng, D. Su, D. Povey, J. Trmal, J. Zhang *et al.*, "Gigaspeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio," *arXiv preprint arXiv:2106.06909*, 2021.

[6] D. Galvez, G. Diamos, J. Ciro, J. F. Cerón, K. Achorn, A. Gopi, D. Kanter, M. Lam, M. Mazumder, and V. J. Reddi, "The People's Speech: A Large-Scale Diverse English Speech Recognition Dataset for Commercial Usage," *arXiv preprint arXiv:2111.09344*, 2021.

[7] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[8] C. Yi, J. Wang, N. Cheng, S. Zhou, and B. Xu, "Applying wav2vec2. 0 to speech recognition in various low-resource languages," *arXiv preprint arXiv:2012.12121*, 2020.

[9] L. Kürzinger, D. Winkelbauer, L. Li, T. Watzel, and G. Rigoll, "CTC-segmentation of large corpora for German end-to-end speech recognition," in *Proc. SPECOM 2020*. Springer, 2020, pp. 267–278.

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

[11] O. Kuchaiev, J. Li, H. Nguyen, O. Hrinchuk, R. Leary, B. Ginsburg, S. Kriman, S. Beliaev, V. Lavrukhin, J. Cook *et al.*, "NeMo: a toolkit for building AI applications using Neural Modules," *arXiv preprint arXiv:1909.09577*, 2019.

[12] S. Majumdar, J. Balam, O. Hrinchuk, V. Lavrukhin, V. Noroozi, and B. Ginsburg, "Citrinet: Closing the gap between non-autoregressive and autoregressive end-to-end models for automatic speech recognition," *arXiv preprint arXiv:2104.01721*, 2021.

[13] M. Del Rio, N. Delworth, R. Westerman, M. Huang, N. Bhandari, J. Palakapilly, Q. McNamara, J. Dong, P. Żelasko, and M. Jetté, "Earnings-21: A Practical Benchmark for ASR in the Wild," in *Proc. Interspeech 2021*, 2021, pp. 3465–3469.

[14] A. Papadopoulos Korfiatis, F. Moramarco, R. Sarac, and A. Savkov, "PriMock57: A dataset of primary care mock consultations," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 588–598.

[15] T. Kim, K. M. Yoo, and S.-g. Lee, "Self-guided contrastive learning for BERT sentence representations," 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. 2528–2540.

[16] T. Parcollet, M. Ravanelli, 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. de Mori, and Y. Bengio, "SpeechBrain: A General-Purpose Speech Toolkit," Mar. 2022, preprint. [Online]. Available: https://hal.science/hal-03601303