TOWARDS REPRESENTATIVE SUBSET SELECTION FOR SELF-SUPERVISED SPEECH RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-supervised speech recognition models require considerable labeled training data for learning high-fidelity representations for Automatic Speech Recognition (ASR) which is computationally demanding and time-consuming, thereby hindering the usage of these models in resource-constrained environments. We consider the task of identifying an optimal subset of data to train self-supervised speech models for ASR. We make a surprising observation that the dataset pruning strategies used in vision tasks for sampling the most informative examples do not perform better than random subset selection on the task of fine-tuning self-supervised ASR. We then present the COWERAGE algorithm for better subset selection in self-supervised ASR, which is based on our finding that ensuring the coverage of examples based on training Word Error Rate (WER) in the early training epochs leads to better generalization performance. Extensive experiments on the wav2vec 2.0 model and TIMIT, Librispeech and LJSpeech datasets show the effectiveness of COWERAGE, with up to 17% relative WER improvement over existing dataset pruning methods and random sampling. We also demonstrate that the coverage of training instances in terms of WER ensures inclusion of phonemically diverse examples which leads to better test accuracy in self-supervised speech recognition models.

1 INTRODUCTION

There has been rapid progress in recent years towards improving speech self-supervised learning (speech SSL) models. Such models learn high-fidelity speech representations using a large amount of untranscribed data and use paired data for fine-tuning on the downstream task of automatic speech recognition (ASR) (Baevski et al., 2020; Hsu et al., 2021). However, a significant amount of labeled training data is used in the fine-tuning step which is computationally demanding and time-consuming (Lai et al., 2021). It is therefore important to consider which training examples are actually important for generalization and whether we can find smaller yet representative subsets of data for fine-tuning speech SSL models. These subsets can eventually aid in training these models in low-resource systems especially compute-restricted environments (e.g., on-device computing), which is presently a significant barrier in democratizing access to these models (Ahmed & Wahed, 2020; Paul et al., 2021). Finally, identifying how example diversity within optimal subsets affects generalization in speech SSL is an important theoretical area requiring further exploration.

The data pruning mechanisms specifically tailored for deep learning models have been studied extensively for standard vision tasks. These methods focus on selecting training examples that are most informative (Toneva et al., 2018; Coleman et al., 2019; Paul et al., 2021; Raju et al., 2021; Karamcheti et al., 2021; Margatina et al., 2021) which has been shown to perform better than the random selection of the training data. The methods for identifying the important examples in these cases are based on scores that are directly derived from the training properties and example difficulty such as the error vector norm (Paul et al., 2021), gradient norm, or the number of times an example is forgotten during training (Toneva et al., 2018). However, no such mechanism has been studied yet for data pruning in speech SSL models.

Studying the impact of the data subset selection on ASR model performance raises a number of questions: Can we identify a scoring method based on the training properties for better dataset pruning in speech SSL without significantly sacrificing the test accuracy? What are the phoneme

distributions of *good* subsets of training data and how do they affect the latent representations within speech SSL models? Can we analyze the training landscape of speech SSL and extract novel insights that can benefit other speech tasks? The answers to these questions will be helpful in constructing smaller datasets that will benefit the paradigm of optimal dataset construction.

We find that in standard datasets for training speech SSL models, sampling only the *hard-to-learn* training examples on the basis of word error rate (WER) does not consistently perform better than random pruning. This is in contrast to data pruning strategies that are frequently used within the deep learning models in vision tasks (Paul et al., 2021; Toneva et al., 2018; Karamcheti et al., 2021). For better data subset selection in training speech SSL models, we propose COWERAGE, an algorithm designed for identifying training examples that are important for better generalization for fine-tuning speech SSL. We find that ensuring the coverage of diverse examples on the basis of *training WER values* in the early training epochs leads to better test accuracy than random pruning or selecting only the most informative (hard-to-learn) examples. The empirical studies show the effectiveness of the COWERAGE algorithm over three other pruning strategies: random selection, top *k* (hardest example selection), and bottom *k* (easiest example selection). To understand the underlying mechanism governing COWERAGE's generalization properties, we establish a connection between the training WER of the examples and their phonemic cover, and find that our algorithm in fact ensures the inclusion of phonemically diverse examples (i.e., examples of both low and high phonemic coverage) without explicitly learning any phoneme-level error model. Finally, we demonstrate that phonemic diversity affects discrete latent representation within speech SSL which leads to performance gains via COWERAGE subset selection.

1.1 OUR CONTRIBUTIONS

- We propose to use the WER of the *individual training examples* as the basis for subset selection algorithms that prune the training data for speech SSL models (Section 3).
- We present COWERAGE, an algorithm for selecting a subset of ASR training data that ensures uniform coverage of training WER values via a bucketing approach (Section 3.3).
- Our empirical evaluation on the `wav2vec2` model (Baevski et al., 2020) and three standard speech datasets TIMIT (Garofolo et al., 1993), Librispeech 10h (Panayotov et al., 2015) and LJSpeech (Ito & Johnson, 2017) show that fine-tuning on the subset selected by COWERAGE performs better on the test set as compared to three other pruning strategies: random, top *k*, and bottom *k* examples (Section 5).
- We study the properties of the subsets selected by COWERAGE by examining the phonemic coverage of training examples. We find that by ensuring the coverage of training WER values, COWERAGE is able to select phonemically diverse examples, which results in a richer training subset (Section 6). Finally, we uncover the relationship between phonemic diversity and the discrete latent representation within speech SSL which allows COWERAGE to perform better than random subset selection and active learning methods (Section 6.1).

2 PRELIMINARIES

Consider a self-supervised model $f(x;\theta)$ ($\theta \in \mathcal{R}^d$) that is pre-trained on a large unlabelled dataset $x \in \mathcal{D}_u$ on some objective $\mathcal{L}_p$. The model obtained after self-supervised pretraining with weights θ_L is then fine-tuned for the downstream task of ASR with another objective $\mathcal{L}_f$ on a labelled dataset $x \in \mathcal{D}_l$ (which is generally smaller than $\mathcal{D}_u$). $\mathcal{D}_l$ consists of transcribed audios (i.e. audio and the corresponding sentence that was uttered). Our goal is to prune $\mathcal{D}_l$ to obtain a subset B_l such that the performance of self-supervised ASR model $f(x;\theta)$ after fine-tuning on B_l is better than random pruning. We only consider pruning $\mathcal{D}_l$ (and not D_u) since we aim to directly evaluate the impact of different subset selection methods on the downstream task of ASR instead of the unsupervised pre-training of speech SSL model. The performance of an ASR model is commonly evaluated via WER, which is computed by aligning the word sequence generated by the ASR system with the actual transcription (containing *N* words) and calculating the sum of substitutions (*S*), insertions (*I*), and deletions (*D*) (Woodard & Nelson, 1982).

$$\text{WER} = \frac{I + D + S}{N} \tag{1}$$

3 METHOD

A number of active learning approaches are based on the inclusion of *informative* training examples in the dataset for deep learning models, i.e., examples with high error during the training epochs. Such examples have been found to have a greater influence on learning how to correctly label the remaining training data and thus are considered more important than examples with low error (*easier* examples). We first quantify the importance of a training example in the context of a self-supervised ASR system to form a baseline for the comparison of different pruning algorithms. The training WER of an example after a few training epochs is representative of the difficulty of that example in being transcribed correctly by an ASR system. Intuitively, a hard-to-learn example will have a higher training WER due to the greater misalignment between the generated word sequence and the actual transcription. We now use the training WER to present three different subset selection strategies for selecting a subset B_l of the training data D_l for fine-tuning a self-supervised speech model on ASR.

3.1 STRATEGY 1: PICKING THE HARDEST *k* EXAMPLES

Algorithm 1 Top *k* Example Selection for Finetuning ASR Model

Input: SSL Pretrained Model f, Dataset D_l, Pruning Fraction p, Training Epoch e
$W \leftarrow$ Finetune f on D_l and compute WER for each example on epoch e
$retainFraction \leftarrow 1 - p$
$retainSize \leftarrow retainFraction * len(D_l)$
$W \leftarrow sortDescending(W)$
$B_l \leftarrow W[0 : retainSize]$

The first approach is to pick the top k training examples, i.e., the ones with the highest WER (Algorithm 1). This replicates the pruning strategy of picking the highest error examples (Paul et al., 2021; Margatina et al., 2021) during training. We first compute the training WER in a particular epoch (WER selection epoch) for all the examples. Then we select examples with the highest WER and perform fine-tuning on this subset. The number of examples selected is determined by the pruning fraction *p*.

3.2 STRATEGY 2: PICKING THE EASIEST *k* EXAMPLES

The second strategy is to pick the bottom k training examples i.e., the ones with the lowest WER (Algorithm 2). This is the inverse of strategy 3.1 and removes the harder-to-learn outliers from the training set in an attempt to retain representative examples.

Algorithm 2 Bottom *k* Example Selection for Finetuning ASR Model

Input: SSL Pretrained Model f, Dataset D_l, Pruning Fraction p, Training Epoch e
$W \leftarrow$ Fine-tune f on D_l and compute WER for each example on epoch e
$retainFraction \leftarrow 1 - p$
$retainSize \leftarrow retainFraction * len(D_l)$
$W \leftarrow sortAscending(W)$
$B_l \leftarrow W[0 : retainSize]$

3.3 STRATEGY 3: COWERAGE SUBSET SELECTION

We now present a novel approach for dataset pruning, which we call COWERAGE, i.e., picking examples to ensure the *coverage* of the training WER. The following claim forms the basis of the COWERAGE algorithm, which we prove later through multiple experiments (Section 5).

Claim 3.1. Ensuring the coverage of training WER values guarantees the inclusion of phonemically diverse examples in the training data.

With COWERAGE, we first compute the training WER for each example in D_l, with the lowest WER as w_l and the highest WER as w_h. We then use a stratified sampling approach of partitioning N total examples from the range $[w_l, w_h]$ into M buckets, with each bucket defined as,

Algorithm 3 COWERAGE Subset Selection for Finetuning ASR Model

Input: SSL Pretrained Model f, Dataset D_l, Pruning Fraction p, Training Epoch e, Bucket Size b
$W \leftarrow$ Finetune f on D_l and compute WER for each example on epoch e
$retainFraction \leftarrow 1 - p$
$B_l \leftarrow EMPTYSET$
$W \leftarrow sortDescending(W)$
$buckets \leftarrow createBuckets(W, size = b)$
for $bucket$ **in** $buckets$ **do**
  $sampleSize \leftarrow retainFraction * b$
  $S \leftarrow randomSample(bucket, sampleSize)$
  $B_l \leftarrow B_l \cup S$

$$S_i = \mathcal{W}\left(w_l + \frac{i-1}{M}\left(w_h - w_l\right), w_l + \frac{i}{M}\left(w_h - w_l\right)\right) \quad i = 1 \ldots n \tag{2}$$

We then use simple random sampling to select k examples uniformly from each bucket,

$$X_{1,\ldots,}X_k \sim \mathcal{U}\left(S_i\right) \tag{3}$$

where k is decided by the fraction of the dataset to be pruned and the size of the bucket. $\mathcal{U}\left(S_i\right)$ denotes the uniform distribution over the set S_i. This stratified sampling method ensures coverage of WER when selecting training examples. The selected subset is used to fine-tune speech SSL model for ASR and the test performance is evaluated through WER (Fig. 1). The overall algorithm is presented in Algorithm 3.

The initial training on the complete training set (step 1 in Fig. 1) needs to be done once to produce the ranking of examples. This ranking can be utilized by a pruning strategy to create an optimal subset (step 2 in Fig. 1). This subset can subsequently be used for downstream fine-tuning of multiple ASR models. This amortizes the initial cost of complete training run across the efficiency improvements achieved via multiple fine-tunings done using the created subset. Sorscher et al. (2022) identify such pruned datasets as *foundation datasets* which can be used for multiple downstream tasks.

3.3.1 COMPARISON TO RANDOM SAMPLING

We now highlight some key differences between random subset selection and COWERAGE.

Claim 3.2. In contrast to the COWERAGE algorithm, random sampling does not ensure selection of examples from the tail WER range.

The proof is presented in Appendix A.1.

Claim 3.3. Subsets selected by COWERAGE have a lower variance of the sample mean of WER than randomly selected samples.

The proof is presented in Appendix A.2.

4 CONFIGURATIONS

4.1 SETUP

We use the `wav2vec2-base` model (Baevski et al., 2020) for our experiments. It consists of a CNN-based encoder that processes the input waveform which is then discretized via the quantization layer and passed to the BERT module where the actual contextual representation is learnt. We select `wav2vec2` which is pretrained on Librispeech 960h with the predictive coding objective. We fine-tune `wav2vec2` for ASR using the Connectionist Temporal Classification (CTC) loss (Graves et al., 2006) on three standard speech datasets: TIMIT (Garofolo et al., 1993), Librispeech 10h (Panayotov et al., 2015) and LJSpeech (Ito & Johnson, 2017). We report WER across multiple pruning fractions and different subset selection methods. Experiments on two other speech SSL models (`HuBERT` and `wav2vec-large`) are presented in Appendix C.2.

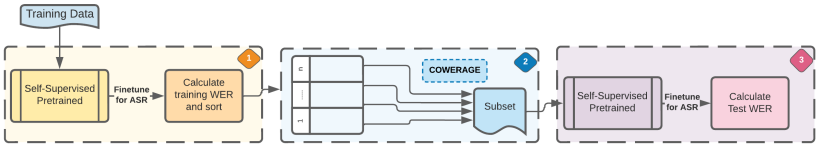

Figure 1: A conceptual representation of the complete flow for fine-tuning a self-supervised model for ASR using a data subset selected by the COWERAGE algorithm. In step (1), we perform fine-tuning on the downstream ASR task using the complete dataset and calculate the training WER on a certain epoch. In step (2), we use COWERAGE to create a data subset by bucketing the training WER and selecting a subset of examples from each bucket. We then use this data in step (3) to fine-tune the complete model and then evaluate on the test dataset.

Table 1: Test WER for the four strategies of pruning the training set evaluated at multiple pruning fractions and different datasets. The training WER in a particular epoch is averaged over 10 runs and then used for a particular pruning strategy. For each result, we do three independent runs and report the mean test WER. The COWERAGE consistently demonstrates the lowest WER at various pruning fractions. WER selection epoch is set to 8 for these experiments (see Section C.1 for ablations).

Dataset	Strategy	Pruning Fraction								
		0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
LJSpeech	Random	0.062	0.065	0.071	0.077	0.085	0.101	0.128	0.160	0.251
	Top K	0.060	0.059	0.064	0.070	0.077	0.085	0.101	0.138	0.238
	Bottom K	0.057	0.058	0.063	0.067	0.070	0.083	0.091	0.110	0.166
	COWERAGE	**0.054**	**0.056**	**0.060**	**0.064**	**0.067**	**0.077**	**0.085**	**0.091**	**0.144**
LS-10h	Random	0.147	0.156	0.168	0.175	0.188	0.210	0.245	0.350	0.360
	Top K	0.143	0.146	0.155	0.158	0.174	0.175	0.198	0.237	0.343
	Bottom K	0.146	0.149	0.159	0.160	0.175	0.178	0.201	0.236	0.336
	COWERAGE	**0.142**	**0.145**	**0.150**	**0.157**	**0.164**	**0.170**	**0.192**	**0.230**	**0.277**
TIMIT	Random	0.457	0.463	0.476	0.483	0.495	0.519	0.606	0.768	1.000
	Top K	**0.453**	0.462	0.486	0.518	0.548	0.591	0.654	0.841	1.000
	Bottom K	0.455	0.468	0.490	0.502	0.517	0.561	0.646	0.819	1.000
	COWERAGE	0.454	**0.460**	**0.461**	**0.470**	**0.474**	**0.516**	**0.587**	**0.754**	1.000

4.2 BASELINE

We consider the baseline experiment of randomly pruning the train split of the dataset on multiple fractions and fine-tuning the ASR model on the generated subset. The performance evaluation is done through WER on the test set.

5 EMPIRICAL EVALUATION

Experiments. We fine-tune `wav2vec2-base` model on the selected dataset and calculate the WER of the training examples over ten independent runs. The training scores (averaged over 10 runs) from a particular epoch are then used to prune the examples through the pruning strategies (3.1, 3.2, 3.3) to generate a subset of training data. The data subsets are then used to fine-tune `wav2vec2` for ASR. The training WER distribution and the subsets of TIMIT, Librispeech and LJSpeech selected through each method are shown in Appendix C.3.

Results. We show the results of pruning experiments via different strategies across multiple pruning fractions in Table 1. For each strategy and pruning fraction, we report the mean WER of three independent runs. The variability across runs is shown in Appendix 11. We observe that for the majority of pruning fractions, COWERAGE subset selection is consistently better than the other three pruning strategies (top k, bottom k, and random pruning) for all the datasets. At higher pruning fractions, the difference between the test WER for COWERAGE and the other pruning strategies

increases, e.g. on the Librispeech-10h dataset with 90% pruning, COWERAGE shows 17% relative WER improvement over Bottom K strategy compared to 5% relative WER improvement at 30% pruning. This observation can also be made for random sampling and is consistent with claim 3.2 where we consider the impact of smaller sample sizes (higher pruning percentages) on the selection of examples from tail WER which subsequently affects test error.

5.1 PHONEME RECOGNITION ON TIMIT

We evaluate the subset selection methods on the task of phoneme recognition with `wav2vec2` on TIMIT dataset and report the phoneme error rate (PER) on the test set (Table 2). COWERAGE consistently demonstrates the lowest PER on all the pruning fractions above 0.2.

Table 2: Phoneme recognition on the TIMIT dataset. We report PER for multiple pruning fractions and different strategies.

Strategy	Pruning Fraction								
	0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
Random	0.124	0.126	0.133	0.139	0.148	0.170	0.230	0.417	1.000
Top K	**0.118**	0.125	0.137	0.152	0.168	0.192	0.244	0.540	1.000
Bottom K	0.122	0.132	0.142	0.156	0.170	0.200	0.282	0.628	1.000
COWERAGE	0.120	**0.123**	**0.133**	**0.135**	**0.145**	**0.147**	**0.211**	**0.218**	1.000

5.2 THE IMPACT OF OFFSET

To identify whether there is another contiguous subset of examples below the ones with the highest WER which can perform better than random pruning, we introduce an offset while selecting the top k training examples, mirroring the protocol presented by Paul et al. (2021).

We compute the training WER for the examples and sort them in ascending order. We then maintain a sliding window from offset k to $k + N$ which keeps N data points but incrementally excludes the training examples with the highest WER. For offset sizes from 0 to 400, we notice a change in the test WER but no single offset size is consistently better than random pruning. An important implication of this finding is that no contiguous subset of training examples picked according to the WER is better than random pruning in the TIMIT speech corpus, contrary to the previous studies on vision datasets that have shown a clear correlation between the top-scoring examples and the accuracy (Paul et al., 2021).

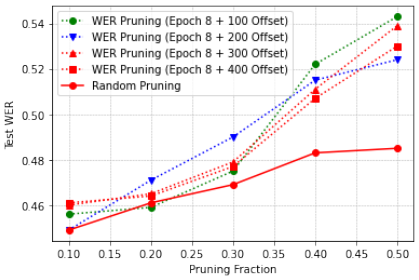

Figure 2: The test WER for the different offsets while picking the top k examples compared over different pruning fractions of the TIMIT dataset. Note that no single offset consistently performs better than random pruning.

5.3 SELECTING WITHIN THE BUCKETS

The strategy proposed in the original COWERAGE algorithm is to sample elements randomly from each bucket. We also evaluate two other strategies: picking the first k examples within each bucket and picking the last k ones, similar to strategies 1 and 2 except that now we are sampling within a particular bucket. The results in Table 3 show that the random selection outperforms other strategies. Additionally, we evaluate the impact of increasing the bucket size on the test WER in Appendix C.5.

5.4 TRAINING TIME FOR SUBSETS

Practically, the choice of pruning fraction can be made according to the intended size of the final dataset under the given time and memory constraints. We conduct an experiment to determine the

Table 3: Test WER for different strategies of picking samples within each bucket for the pruning fraction of 0.5 and WER selection epoch 8.

	Top k	Bottom k	Random
Test WER	0.489 ± 0.002	0.496 ± 0.002	$\mathbf{0.474 \pm 0.005}$

total steps required for convergence and the real training time for `wav2vec2` on TIMIT. The results are shown in Table 4 (for a constant learning rate). We report the real training time for the pruned datasets as a fraction of the training time for the complete dataset (x) for relative comparison. There is a significant reduction in training time for higher pruning fractions.

Table 4: Steps required for convergence and training time for `wav2vec2` on TIMIT for different pruning fractions

Pruning Fraction	0.8	0.6	0.4	0.2	0
Steps required for convergence	1600	2200	2600	3000	3350
Training time	$0.52x$	$0.77x$	$0.90x$	$0.96x$	x

6 Connection to Phonemes

To understand why COWERAGE performs better than other pruning strategies, it is important to find out how does the phoneme distribution of training examples vary with the training error during fine-tuning of the self-supervised speech recognition models. We now perform empirical analysis to verify claim 3.1. For this analysis, we select the standard TIMIT dataset as it contains time-aligned, hand-verified phonetic and word transcriptions for each training example.

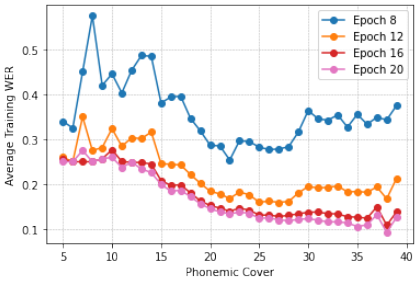

Figure 3: The training WER and the phonemic cover of examples in TIMIT dataset (without pruning) compared over multiple training epochs. The WER is computed by averaging the training scores of the examples with the same phonemic cover. The training scores for each training example and a particular epoch are computed by averaging over 10 runs.

We first record the training WER of each training example in the TIMIT dataset over 10 runs and average it. Then, we compute the total number of unique phonemes in each example, which we call the *phonemic cover*. Subsequently, we group together the training examples with same phonemic cover and calculate the average training WER for each group (Fig. 3). In the earlier training epochs, the examples with a relatively low (< 17) or a high (> 28) phonemic cover have a greater WER (blue line in Fig. 3) as compared to the examples with a moderate number of phonemes ($17 \leq phonemicCover \leq 28$). In the later epochs ($\geq 12$), the inverse relationship between the training WER and the phonemic cover becomes more evident; the examples with a greater number of distinct phonemes have a lower training WER and vice versa.

Significance. This relationship between the training WER and the phonemic cover has several implications. Firstly, it demonstrates that there is a sizable population of sentences with a low phonemic cover that are harder to learn and hence represent a high training WER. Similarly, there are many low WER sentences with a high phonemic cover (examples are presented in Appendix C.9). More importantly, this experiment validates our claim that ensuring the coverage of training WER values in a particular subset leads to the inclusion of phonemically diverse training examples *without* explicitly learning any phoneme-level error model. This is beneficial as accurate phonetic data is not available for the majority of 7000 spoken languages (Billington et al., 2021). In contrast, any method that directly ensures phoneme diversity requires an accurate phonetic transcription beforehand, which is a resource-intensive process requiring manual labeling by linguists.

To verify if the difference between the phoneme distributions of the examples within the COWERAGE subset and the other two strategies (top *k* and bottom *k*) is statistically significant, we conduct the Mann-Whitney U test, a non-parametric test, at a significance level of .01. The differences are found to be statistically significant, with a *p*-value $< .01$. The results are shown in Table 5.

Table 5: The statistical significance of the difference between the phoneme distribution of the examples within the COWERAGE subset and the other two strategies (top *k* and bottom *k*). * Indicates the difference is significant ($\alpha = 0.01$); MWU: Mann-Whitney U.

	MWU	*p*-value
Top *k* vs COWERAGE	2146027.5	$4.28 * 10^{-31}*$
Bottom *k* vs COWERAGE	2229653.0	$1.58 * 10^{-22}*$

6.1 PHONEMIC DIVERSITY AND LATENT REPRESENTATION IN SPEECH SSL

How does phonemic diversity impact the discrete latent speech representations within self-supervised speech recognition models? More specifically, we consider the latent representation ($\mathbf{q}_t$) learned by the quantizer within `wav2vec2` for different phonemes. Baevski et al. (2020) analyze the conditional probability $P\left(phoneme \mid \mathbf{q}_t\right)$ for each of the 39 phonemes in the TIMIT train set by computing the co-occurence between the phonemes and speech latents (see Appendix D of Baevski et al. (2020)). They demonstrate that different discrete latents specialize in different phonetic sounds in `wav2vec2` model. Given this observation, we hypothesize that the performance gains for COWERAGE are due to the greater phonemic diversity which enables a more robust latent representation of each phoneme in `wav2vec2`. This view is supported by the results in Table 1 which demonstrate bigger gains in test WER for higher pruning fractions in COWERAGE. We conjecture that this is due to greater example diversity provided by COWERAGE, and lack of representation of examples from the tail WER range in case of other approaches.

7 RELATED WORK

Devising strategies for data pruning and constructing optimal subsets is a recent topic of interest in the area of optimization and active learning (Dong et al., 2019; Kaushal et al., 2019; Saadatfar et al., 2020; Durga et al., 2021; Kothawade et al., 2021; Killamsetty et al., 2021; Kothyari et al., 2021; Ahia et al., 2021). A few studies have examined the training landscape for drawing clues about the optimal subset creation (Toneva et al., 2018; Agarwal et al., 2020; Baldock et al., 2021; Paul et al., 2021; Schirrmeister et al., 2022). Toneva et al. (2018) observe the number of times the training examples are 'forgotten' during training and find that rarely forgotten examples can be effectively eliminated from the training subset without affecting the generalization accuracy. Paul et al. (2021) evaluate the impact of static data pruning on the performance on standard vision datasets (e.g., CIFAR-10 and CIFAR-100) and models (ResNet). They use the gradient norm (GraNd) and the error norm (EL2N) for removing the 'easy' training examples and pruning a significant chunk of the dataset without affecting the generalization error. The authors observe that the local information in the early training epochs is a strong indicator of the importance of training examples and thus can be used to effectively select a good subset of training data. This is consistent with our observation regarding WER selection epoch.

The work on coresets (Tolochinsky & Feldman, 2018; Huang et al., 2021; Jiang et al., 2021; Jubran et al., 2021; Mirzasoleiman et al., 2020) is also being actively researched in the regime of optimization. It refers to the strategy of constructing a subset by sampling from the original dataset in a manner that an approximate solution with a bounded error can be reached when an algorithm is run on it. They have been shown to work on particular machine learning approaches, and relatively fewer studies have demonstrated their application in deep learning as these algorithms require the problem to demonstrate a special structure such as convexity (Paul et al., 2021).

Although sampling hard-to-learn examples has been a popular choice for data pruning in deep learning models, it appears to work on a limited set of tasks that share certain properties. A study on visual question answering (VQA) (Karamcheti et al., 2021) demonstrates that the active learning approaches that prefer picking the *harder* examples do not outperform random pruning on the VQA task across

multiple models and datasets. The authors demonstrate the role of collective outliers (Han et al., 2011) in degrading the generalization performance and find out that the preference for selecting these harder-to-learn outliers by the active learning methods is the cause of poor improvements in efficiency as compared to random sampling.

The existing work on active learning and data pruning for ASR systems emphasize the importance of ensuring phonemically rich text and higher coverage of words (Wu et al., 2007; Ni et al., 2015a; Wei et al., 2014; Mendonça et al., 2014; Ni et al., 2015b;a; 2016). An early study (Wu et al., 2007) demonstrates that selecting a subset that is sampled uniformly across phonemes and words is more effective than random sampling. A subsequent work (Wei et al., 2014) proposes a method for selecting the data by maximizing a constrained sub-modular function. The results show the possibility of a significant reduction of the training data when using acoustic models based on Gaussian mixture models. In ASR, active learning aims to select the most informative utterances to be transcribed from a large amount of un-transcribed utterances. In contrast, our core objective is to construct an optimal data subset by selecting the informative and representative examples from a *fully labeled* dataset i.e. the examples for which audios and the reference transcriptions are available.

The majority of these existing approaches have focused on the earlier ASR systems instead of the Deep Neural Network (DNN) based models. Although model pruning has been explored for self-supervised and other ASR models (Lai et al., 2021; Wu et al., 2021; Zhen et al., 2021), data subset selection for fine-tuning self-supervised ASR systems has only been explored in the context of personalization for accented speakers (Awasthi et al., 2021). A phoneme-level error model is proposed which selects sentences that yield a lower test WER as compared to random sentence selection. The common idea in these works is to construct data subsets according to a certain phonemic distribution which works better than random pruning. In contrast, our COWERAGE algorithm has the advantage that no complex, dataset-specific phoneme-level error model needs to be learned which constructs phonemically diverse subsets. Instead, just the training WER can be used to devise a dataset agnostic strategy for pruning that performs better than random selection. Additionally, to the best of our knowledge, this is the first study that considers the data pruning in the context of self-supervised speech recognition models.

8 LIMITATIONS AND FUTURE WORK

We now present some limitations of our work and discuss potential directions for future work. While we designed our approach to be dataset agnostic and applicable to different distributions of training WER, it remains to be empirically evaluated whether our methodology generalizes to other types of datasets for ASR e.g. different languages, nosier data etc. Moreover, it will be useful to apply subset selection to other speech tasks including text-to-speech, keyword spotting and speaker verification.

9 CONCLUSION

In this work, we proposed COWERAGE, a new method for pruning data for self-supervised automatic speech recognition, which relies on sampling data in a way that ensures coverage of training WER values. An evaluation on `wav2vec2` and three datasets show that COWERAGE performs better than random selection and other data pruning strategies that select harder-to-learn or easier-to-learn examples. We unveil the connection between the training word error rate and the phonemic cover of training examples across multiple training epochs and analyze the pruning results through this lens. Finally, we show that COWERAGE outperforms other subset selection strategies as it ensures phonemic diversity within the training examples by directly utilizing the training WER of speech SSL models.

10 REPRODUCIBILITY STATEMENT

We include the complete proofs of the claims in Appendix A. We also describe the implementation details including datasets, hyperparameters and training procedures in Appendix B. The source code is submitted as part of the supplementary material.

REFERENCES

Chirag Agarwal, Daniel D'souza, and Sara Hooker. Estimating example difficulty using variance of gradients. *arXiv preprint arXiv:2008.11600*, 2020.

Orevaoghene Ahia, Julia Kreutzer, and Sara Hooker. The low-resource double bind: An empirical study of pruning for low-resource machine translation. *arXiv preprint arXiv:2110.03036*, 2021.

Nur Ahmed and Muntasir Wahed. The de-democratization of ai: Deep learning and the compute divide in artificial intelligence research. *arXiv preprint arXiv:2010.15581*, 2020.

Abhijeet Awasthi, Aman Kansal, Sunita Sarawagi, and Preethi Jyothi. Error-driven fixed-budget asr personalization for accented speakers. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7033–7037. IEEE, 2021.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*, 2020.

Robert JN Baldock, Hartmut Maennel, and Behnam Neyshabur. Deep learning through the lens of example difficulty. *arXiv preprint arXiv:2106.09647*, 2021.

Rosey Billington, Hywel Stoakes, and Nick Thieberger. The Pacific Expansion: Optimizing Phonetic Transcription of Archival Corpora. In *Proc. Interspeech 2021*, pp. 4029–4033, 2021. doi: 10.21437/Interspeech.2021-2167.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. *arXiv preprint arXiv:1906.11829*, 2019.

Luobing Dong, Qiumin Guo, and Weili Wu. Speech corpora subset selection based on time-continuous utterances features. *Journal of Combinatorial Optimization*, 37(4):1237–1248, 2019.

S Durga, Rishabh Iyer, Ganesh Ramakrishnan, and Abir De. Training data subset selection for regression with controlled generalization error. In *International Conference on Machine Learning*, pp. 9202–9212. PMLR, 2021.

John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon technical report n*, 93:27403, 1993.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pp. 369–376, 2006.

Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *arXiv preprint arXiv:2106.07447*, 2021.

Lingxiao Huang, K Sudhir, and Nisheeth Vishnoi. Coresets for time series clustering. *Advances in Neural Information Processing Systems*, 34, 2021.

Keith Ito and Linda Johnson. The lj speech dataset, 2017.

Shaofeng Jiang, Robert Krauthgamer, Xuan Wu, et al. Coresets for clustering with missing values. *Advances in Neural Information Processing Systems*, 34, 2021.

Ibrahim Jubran, Ernesto Evgeniy Sanches Shayda, Ilan Newman, and Dan Feldman. Coresets for decision trees of signals. *Advances in Neural Information Processing Systems*, 34, 2021.

Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei, and Christopher D Manning. Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. *arXiv preprint arXiv:2107.02331*, 2021.

Vishal Kaushal, Rishabh Iyer, Suraj Kothawade, Rohan Mahadev, Khoshrav Doctor, and Ganesh Ramakrishnan. Learning from less data: A unified data subset selection and active learning framework for computer vision. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1289–1299. IEEE, 2019.

Krishnateja Killamsetty, Durga Sivasubramanian, Baharan Mirzasoleiman, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: A gradient matching based data subset selection for efficient learning. *arXiv preprint arXiv:2103.00123*, 2021.

Suraj Kothawade, Nathan Beck, Krishnateja Killamsetty, and Rishabh Iyer. Similar: Submodular information measures based active learning in realistic scenarios. *Advances in Neural Information Processing Systems*, 34, 2021.

Mayank Kothyari, Anmol Reddy Mekala, Rishabh Iyer, Ganesh Ramakrishnan, and Preethi Jyothi. Personalizing asr with limited data using targeted subset selection. *arXiv preprint arXiv:2110.04908*, 2021.

Cheng-I Jeff Lai, Yang Zhang, Alexander H Liu, Shiyu Chang, Yi-Lun Liao, Yung-Sung Chuang, Kaizhi Qian, Sameer Khurana, David Cox, and James Glass. Parp: Prune, adjust and re-prune for self-supervised speech recognition. *arXiv preprint arXiv:2106.05933*, 2021.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 650–663, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

Gustavo Mendonça, Sara Candeias, Fernando Perdigao, Christopher Shulby, Rean Toniazzo, Aldebaro Klautau, and Sandra Aluísio. A method for the extraction of phonetically-rich triphone sentences. In *2014 International Telecommunications Symposium (ITS)*, pp. 1–5. IEEE, 2014.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pp. 6950–6960. PMLR, 2020.

Chongjia Ni, Cheung-Chi Leung, Lei Wang, Nancy F Chen, and Bin Ma. Unsupervised data selection and word-morph mixed language model for tamil low-resource keyword search. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4714–4718. IEEE, 2015a.

Chongjia Ni, Lei Wang, Haibo Liu, Cheung-Chi Leung, Li Lu, and Bin Ma. Submodular data selection with acoustic and phonetic features for automatic speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4629–4633. IEEE, 2015b.

Chongjia Ni, Cheung-Chi Leung, Lei Wang, Haibo Liu, Feng Rao, Li Lu, Nancy F Chen, Bin Ma, and Haizhou Li. Cross-lingual deep neural network based submodular unbiased data selection for low-resource keyword search. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6015–6019. IEEE, 2016.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210. IEEE, 2015.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34, 2021.

Ravi S Raju, Kyle Daruwalla, and Mikko Lipasti. Accelerating deep learning with dynamic data pruning. *arXiv preprint arXiv:2111.12621*, 2021.

Hamid Saadatfar, Samiyeh Khosravi, Javad Hassannataj Joloudari, Amir Mosavi, and Shahaboddin Shamshirband. A new k-nearest neighbors classifier for big data based on efficient data pruning. *Mathematics*, 8(2):286, 2020.

Robin Tibor Schirrmeister, Rosanne Liu, Sara Hooker, and Tonio Ball. When less is more: Simplifying inputs aids neural network understanding. *arXiv preprint arXiv:2201.05610*, 2022.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *arXiv preprint arXiv:2206.14486*, 2022.

Elad Tolochinsky and Dan Feldman. Coresets for monotonic functions with applications to deep learning. *CoRR, abs/1802.07382*, 2018.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

Kai Wei, Yuzong Liu, Katrin Kirchhoff, Chris Bartels, and Jeff Bilmes. Submodular subset selection for large-scale speech training data. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3311–3315. IEEE, 2014.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

JP Woodard and JT Nelson. An information theoretic measure of speech recognition performance. In *Workshop on standardisation for speech I/O technology, Naval Air Development Center, Warminster, PA*, 1982.

Yi Wu, Rong Zhang, and Alexander Rudnicky. Data selection for speech recognition. In *2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, pp. 562–565. IEEE, 2007.

Zhaofeng Wu, Ding Zhao, Qiao Liang, Jiahui Yu, Anmol Gulati, and Ruoming Pang. Dynamic sparsity neural networks for automatic speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6014–6018. IEEE, 2021.

Kai Zhen, Hieu Duy Nguyen, Feng-Ju Chang, Athanasios Mouchtaris, and Ariya Rastrow. Sparsification via compressed sensing for automatic speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6009–6013. IEEE, 2021.

A PROOFS

A.1 CLAIM 3.2

Claim 3.2 In contrast to the COWERAGE algorithm, random sampling does not ensure selection of examples from the tail WER range.

Proof. We consider the probability of randomly selecting an example WER (w) that is at least at a distance of k standard deviation σ from the mean WER. By Chebyshev's inequality: $\Pr(|X - \bar{W}| \geq k\sigma) \leq \frac{1}{k^2} = p$, which demonstrates that increasing the WER boundary w (and hence k) decreases the probability of randomly selecting a sample with WER greater than w. Note that the bound is inverse *quadratic*. We now consider the probability of having at least one sample with a WER greater w when we independently draw n samples from the training WER distribution. This is a complement of the event "no sample having a WER greater than w in n draws" which is $(1-p)^n$, and hence the event of interest has the probability upper bound $1 - (1-p)^n = 1 - (1 - \frac{1}{k^2})^n$. This demonstrates that decreasing the sample size and increasing the pruning percentage reduces the probability of selecting a tail WER example. In contrast, for COWERAGE, the probability of selecting at least one example with a WER greater than $\bar{W} + k\sigma$ is $Pr(|S_i| > 0) = q$, where S_i is a tail bucket with the WER range (a, b) such that $a \geq \bar{W} + k\sigma$ and $b > a$. This probability (q) approaches 1 if we consider an appropriately large bucket size satisfying the range (a, b), and hence COWERAGE ensures selection of examples from the tail WER range.

□

A.2 CLAIM 3.3

Claim 3.3 Subsets selected by COWERAGE have a lower variance of the sample mean of WER than randomly selected samples.

Proof. We first consider the variance of samples selected by COWERAGE. Let S_{ij} be the sample i from bucket S_j. The average WER in bucket j is $\bar{W}_j = \frac{\sum_i S_{ij}}{k}$, variance in bucket j is σ_j^2 and the overall average is $\bar{W} = \frac{\sum_j \bar{W}_j}{M}$. The variance is,

$$\mathrm{Var}_{\text{COWERAGE}}[\bar{W}] = \frac{\sum_j \mathrm{Var}\left[\bar{W}_j\right]}{M^2} = \frac{\sum_j \sigma_j^2}{M^2 k} = \frac{\sum_j \sigma_j^2}{MN}. \tag{4}$$

Now we consider the variance of a simple random sample. $\mathrm{Var}[\bar{W}] = \frac{\sigma^2}{N}$ with $\sigma^2 = \mathbb{E}\left[W^2\right] - \mu^2$. Considering the contribution from each bucket in the random sample, we can specify $\sigma^2 = \frac{\sum_j \mathbb{E}[S_j]}{M} - \mu^2 = \frac{\sum_j \left(\mu_j^2 + \sigma_j^2\right)}{M} - \mu^2 = \frac{\sum_j \left((\mu_j - \mu)^2 + \sigma_j^2\right)}{M}$. Thus,

$$\mathrm{Var}_{\text{RANDOM}}[\bar{W}] = \frac{\sum_j \left((\mu_j - \mu)^2 + \sigma_j^2\right)}{MN} \tag{5}$$

Comparing equation 4 and equation 5, $\mathrm{Var}_{\text{RANDOM}}[\bar{W}] \geq \mathrm{Var}_{\text{COWERAGE}}[\bar{W}]$ and the result follows. □

B IMPLEMENTATION DETAILS

B.1 RESOURCES

We use a single 80GB NVIDIA A100 GPU for running all the experiments on the cloud. In this setting, the standard `wav2vec2-base` finetuning step (single run) on multiple pruning fractions took ≈ 1.25 GPU hours for the TIMIT dataset, ≈ 6 GPU hours for LJSpeech dataset, and ≈ 5.5 GPU hours for Librispeech 10h dataset. The total project (from the early experiments to the final results) consumed about 2200 GPU hours.

B.2 CODE AND LICENSES

We release our code under the MIT license. All the data pruning strategies are implemented in Python, and the resulting subsets are used to fine-tune `wav2vec2`. The publicly available HuggingFace (Wolf et al., 2019) implementation [1] of `wav2vec2-base` model[2] is used which is based on the standard `wav2vec2-base-960h` fairseq implementation[3]. The HuggingFace transformers repo is available under the Apache License 2.0 license and the fairseq repo is available under the MIT license.

B.3 DATA

TIMIT. We use the full TIMIT dataset (Garofolo et al., 1993) with predefined training and test sets. The training set contains 4620 examples and the test set contains 1680 examples. TIMIT is available under the LDC User Agreement for Non-Members.
Librispeech. We use Librispeech 10h finetuning split (Panayotov et al., 2015) and the complete predefined test split. Librispeech is available under the CC BY 4.0 license.
LJSpeech. We use the complete LJSpeech data (Ito & Johnson, 2017) with predefined training and test sets. LJSpeech is available under the public domain license.

B.4 TRAINING

In all experiments, `wav2vec2-base` is fine-tuned with a batch size = 8, epochs = 20, mean ctc-loss-reduction, gradient checkpointing and FP16 training. We use a data collator to dynamically pad the inputs. For calculating the WER for each training example, we run a computation step after each epoch and record the WER. The training WER in each epoch is averaged over 10 runs and then used for a particular pruning strategy. For each test WER reported, we do three separate runs with independent model initialization. A bucket size of 100 is chosen for the COWERAGE strategy, which is sufficiently small enough to ensure the selection of representative examples for different pruning fractions.

[1] https://github.com/huggingface/transformers
[2] https://huggingface.co/facebook/wav2vec2-base-960h
[3] https://github.com/pytorch/fairseq/blob/main/examples/wav2vec/README.md

C ADDITIONAL EXPERIMENTS

C.1 WER SELECTION EPOCH

An important hyperparameter in the COWERAGE algorithm is the epoch at which the training WER is computed for individual examples and then used for pruning i.e. the WER selection epoch. We evaluate the effect of different selection epochs on the final test WER (Table 6) in TIMIT and observe that the training WER in the early training epochs can be reliably used for ranking the examples and applying a particular pruning strategy. Hence, we select WSE = 8 for the final results in Table 1. Note that COWERAGE consistently demonstrates a lower WER than other strategies on *all epochs* that we test (8, 12, 16, 20) for the majority of pruning fractions ($0.2 - 0.9$) across all the datasets (TIMIT, LS-10h, LJSpeech). This suggests that the selection of a reasonable WSE can usually be made with less than five distinct epoch values while still achieving better results than the other strategies.

Table 6: Test WER for the four strategies of pruning the training set evaluated at multiple pruning fractions and different training WER selection epochs. The training WER in a particular selection epoch is averaged over 10 runs and then used for a particular pruning strategy. For each result, we do three independent runs and report the mean test WER. COWERAGE consistently demonstrates the lowest WER at various pruning fractions and selection epochs. WSE: WER Selection Epoch.

WSE	Strategy	Pruning Fraction								
		0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
8	Random	0.457	0.463	0.476	0.483	0.495	0.519	0.606	0.768	1.000
	Top K	**0.453**	0.462	0.486	0.518	0.548	0.591	0.654	0.841	1.000
	Bottom K	0.455	0.468	0.490	0.502	0.517	0.561	0.646	0.819	1.000
	COWERAGE	0.454	**0.460**	**0.461**	**0.470**	**0.474**	**0.516**	**0.587**	**0.754**	1.000
12	Random	0.457	0.463	0.476	0.483	0.495	0.519	0.606	0.768	1.000
	Top K	**0.450**	0.455	0.500	0.524	0.565	0.583	0.676	0.862	1.000
	Bottom K	0.460	0.458	0.484	0.503	0.511	0.550	0.648	0.873	1.000
	COWERAGE	0.456	**0.461**	**0.461**	**0.479**	**0.491**	**0.501**	**0.568**	**0.741**	1.000
16	Random	0.457	0.463	0.476	0.483	0.495	0.519	0.606	0.768	1.000
	Top K	**0.453**	0.470	0.483	0.560	0.567	0.605	0.687	0.965	1.000
	Bottom K	0.457	0.457	0.484	0.486	0.508	0.541	0.640	0.903	1.000
	COWERAGE	0.455	**0.462**	**0.469**	**0.473**	**0.482**	**0.512**	**0.559**	**0.690**	1.000
20	Random	0.457	0.463	0.476	0.483	0.495	0.519	0.606	0.768	1.000
	Top K	0.458	0.502	0.505	0.558	0.586	0.621	0.714	0.855	1.000
	Bottom K	**0.452**	0.462	0.478	0.488	0.508	0.541	0.648	0.764	1.000
	COWERAGE	0.454	**0.457**	**0.464**	**0.474**	**0.495**	**0.510**	**0.573**	**0.761**	1.000

C.2 OTHER MODELS

We conduct additional experiments on two additional models to demonstrate method robustness:

- `wav2vec2-large` on TIMIT
- `HuBERT` on TIMIT

The results are shown in the Table 7. COWERAGE consistently demonstrates the lowest WER at various pruning fractions.

C.3 TRAINING WER DISTRIBUTION

We compare the distribution of the training WER for TIMIT (Fig. 4), Librispeech 10h (Fig. 5) and LJSpeech (Fig. 6) and show the subsets selected through Top K, Bottom K and COWERAGE subset selection on 50% pruning percentage. We notice significant differences in the training WER distribution for the three datasets which highlights that the example difficulty (measured by WER) is

Table 7: Test WER for the four strategies evaluated at multiple pruning fractions on different models with the TIMIT dataset. WER selection epoch is set to 8 for these experiments.

Model	Strategy	Pruning Fraction			
		0.2	**0.4**	**0.6**	**0.8**
`wav2vec-large`	Random	0.205	0.215	0.243	0.295
	Top K	0.300	0.352	0.420	0.502
	Bottom K	0.207	0.361	0.422	0.540
	COWERAGE	**0.192**	**0.198**	**0.209**	**0.261**
`HuBERT`	Random	0.276	0.306	0.390	0.472
	Top K	0.267	0.336	0.534	0.752
	Bottom K	0.283	0.372	0.383	0.625
	COWERAGE	**0.259**	**0.285**	**0.359**	**0.441**

a property of the dataset. Moreover, since COWERAGE performs better than other subset selection methods across multiple datasets, we hypothesize that the our proposed method is dataset-agnostic and can perform well with different training WER distributions.

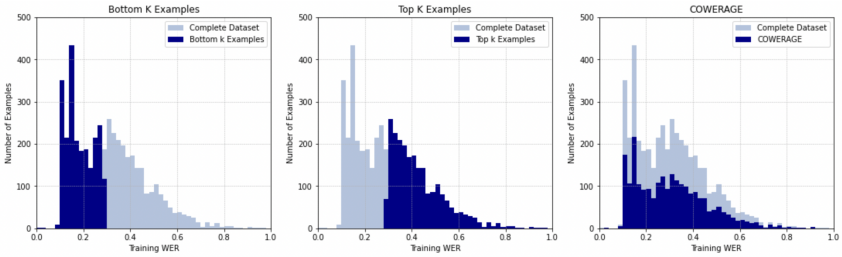

Figure 4: The subset of the TIMIT training data selected by each of the three strategies: bottom k (*left*), top k (*middle*) and COWERAGE (*right*). The pruning fraction is set to 0.5 and the WER selection epoch is 8.

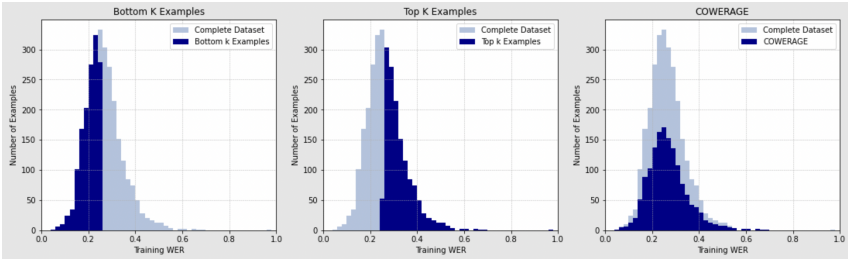

Figure 5: The subset of the Librispeech 10h training data selected by each of the three strategies: bottom k (*left*), top k (*middle*) and COWERAGE (*right*). The pruning fraction is set to 0.5 and the WER selection epoch is 8.

C.4 TRAINING LANDSCAPE

We now compare the training landscape for the three strategies discussed. We create four subsets of data at the pruning fraction of 0.7 and plot the training WER for each of the four approaches (Fig. 7). By examining the outlier behavior and the width of the box plots (25th to 75th percentile), we find that COWERAGE subset selection is actually picking the *moderately hard* and *representative* examples instead of just the *very hard* but *rare* examples. This is in agreement with the plots in Fig. 8 which shows that COWERAGE is able to outperform the other two approaches on unseen data by achieving a significantly lower test WER on the 25th to 50th percentile of examples.

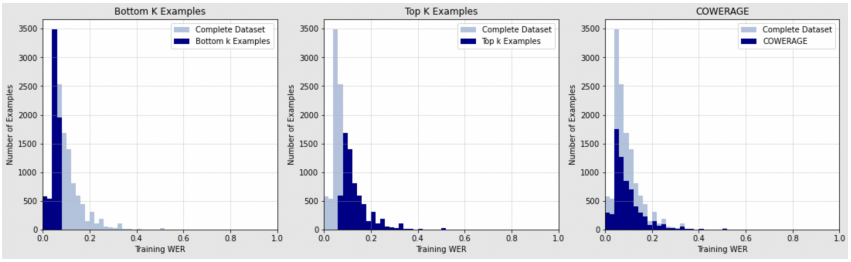

Figure 6: The subset of the LJspeech 24h training data selected by each of the three strategies: bottom k (*left*), top k (*middle*) and COWERAGE (*right*). The pruning fraction is set to 0.5 and the WER selection epoch is 8.

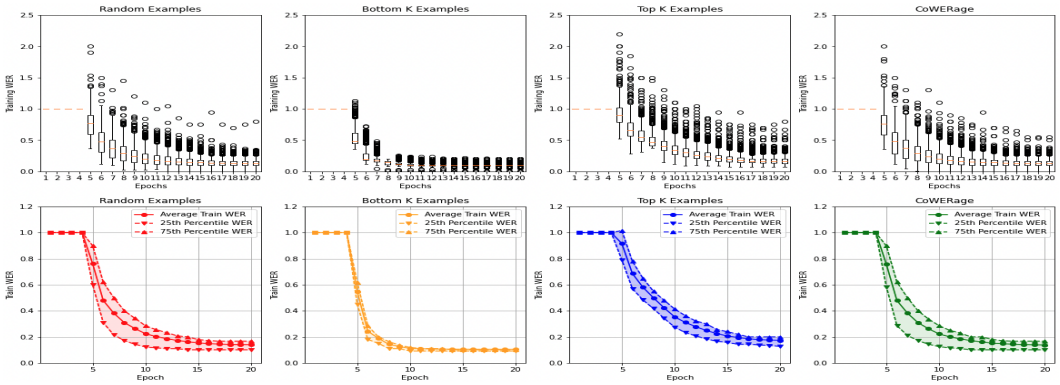

Figure 7: The training trajectories of the examples in TIMIT selected by picking the random examples (*first column*), bottom k examples (*second column*), top k examples (*third column*), and via COWERAGE subset selection (*fourth column*). For each epoch, we show the box plot of the distribution of the word error rate of training examples that indicates the mean, quartiles, and outliers.

C.5 SELECTING THE NUMBER OF BUCKETS

We conduct an experiment with different bucket sizes on `wav2vec2` and TIMIT with 0.5 pruning fraction. The results are shown in Table 8. Our evaluation shows that increasing the bucket size beyond a certain threshold provides diminishing returns in performance. Increasing the bucket size from 50 to 100 yielded 2.7% reduction in WER whereas increasing it from 100 to 500 resulted in only a 0.2% reduction in WER.

Table 8: Test WER for `wav2vec2` on TIMIT for different number of buckets in the COWERAGE algorithm

Number of Buckets	1	10	50	100	500
Test WER	0.494	0.492	0.488	0.474	0.473

Choosing 100 buckets in the COWERAGE algorithm provided robust performance across a wide range of dataset sizes, which ranged from 4620 examples in TIMIT to more than 10,000 examples in LJSpeech. The number of buckets can be increased further but it should be no greater than $pruningFraction * datasetSize$.

C.6 WER WITHOUT PRUNING

We report the original WER when finetuning `wav2vec-base` on the full datasets (i.e., pruning fraction 0.0) in Table 9.

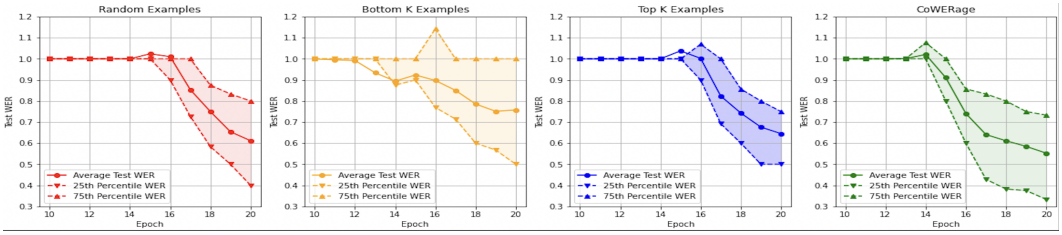

Figure 8: The WER for test examples in TIMIT over multiple training epochs for four different strategies: random pruning (*first column*), picking bottom k examples (*second column*), picking top k examples (*third column*), and picking examples via COWERAGE subset selection (*fourth column*). The pruning fraction is set to 0.7. The solid lines indicate the average test WER computed at each epoch, and the shaded region extends from the 25th to the 75th percentile of test examples WER.

Table 9: Test WER for `wav2vec2` on complete datasets.

Dataset	TIMIT	LS-10h	LJSpeech
WER without pruning	0.449	0.140	0.052

C.7 LENGTH AND PHONEMES

In this section, we examine the relationship between length and the training WER and conduct the same experiment from Section 6 but now with the length instead of the phonemic cover. The results are shown in Figure 9. The overall inverse relationship is similar to the one in Figure 3 but is noisier. We notice that there are shorter and longer sentences with a high training WER in the earlier training epochs. If we bucket the examples by length, each bucket has a higher variance of WER values than the phoneme experiment in Figure 3. We also evaluate a variant of COWERAGE that selects examples on the basis of their character length instead of WER which demonstrates that WER sampling is a better subset selection strategy than length sampling (Table 10).

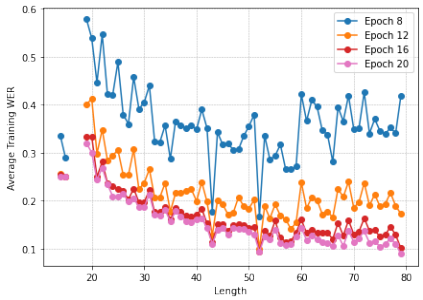

Figure 9: The training WER and the length of the examples (total number of characters) in TIMIT dataset compared over multiple training epochs. The WER is computed by averaging the training scores of the examples with the same length.

Table 10: Test WER for a variant of COWERAGE that selects examples based on their length instead of WER.

Model	Strategy	Pruning Fraction			
		0.2	**0.4**	**0.6**	**0.8**
`HuBERT`	COWERAGE (Length)	0.207	0.361	0.422	0.540
	COWERAGE (WER)	**0.259**	**0.285**	**0.359**	**0.441**

C.8 STANDARD DEVIATION FOR TEST WER ON TIMIT

Table 11: The standard deviation for the test WER on TIMIT dataset presented in Table. 6

WSE	Strategy	Pruning Fraction								
		0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
8	Random	± 0.003	± 0.001	± 0.005	± 0.004	± 0.002	± 0.003	± 0.025	± 0.033	± 0
	Top K	± 0.001	± 0.009	± 0.007	± 0.001	± 0.010	± 0.010	± 0.001	± 0.020	± 0
	Bottom K	± 0.002	± 0.001	± 0.002	± 0.001	± 0.002	± 0.002	± 0.009	± 0.023	± 0
	COWERAGE	± 0.001	± 0.002	± 0.006	± 0.005	± 0.016	± 0.002	± 0.011	± 0.029	± 0
12	Random	± 0.003	± 0.001	± 0.005	± 0.004	± 0.002	± 0.003	± 0.025	± 0.033	± 0
	Top K	± 0.004	± 0.005	± 0.001	± 0.010	± 0.012	± 0.012	± 0.009	± 0.038	± 0
	Bottom K	± 0.004	± 0.006	± 0.004	± 0.001	± 0.004	± 0.003	± 0.007	± 0.025	± 0
	COWERAGE	± 0.002	± 0.001	± 0.006	± 0.007	± 0.002	± 0.002	± 0.004	± 0.030	± 0
16	Random	± 0.003	± 0.001	± 0.005	± 0.004	± 0.002	± 0.003	± 0.025	± 0.033	± 0
	Top K	± 0.002	± 0.003	± 0.003	± 0.011	± 0.005	± 0.009	± 0.008	± 0.001	± 0
	Bottom K	± 0.001	± 0.005	± 0.002	± 0.002	± 0.003	± 0.003	± 0.012	± 0.030	± 0
	COWERAGE	± 0.003	± 0.001	± 0.001	± 0.003	± 0.003	± 0.008	± 0.001	± 0.006	± 0
20	Random	± 0.003	± 0.001	± 0.005	± 0.004	± 0.002	± 0.003	± 0.025	± 0.033	± 0
	Top K	± 0.002	± 0.002	± 0.002	± 0.005	± 0.008	± 0.006	± 0.022	± 0.039	± 0
	Bottom K	± 0.001	± 0.001	± 0.006	± 0.001	± 0.001	± 0.010	± 0.006	± 0.072	± 0
	COWERAGE	± 0.004	± 0.001	± 0.001	± 0.003	± 0.005	± 0.001	± 0.009	± 0.042	± 0

C.9 EXAMPLES

Table 12: Training examples in the TIMIT dataset and their training WER on `wav2vec2` along with the phonemic cover (PC). The training WER is calculated by averaging 10 runs.

WER	Text	Phonemes	PC
0.63	Twelve o'clock level.	(t-w-eh-l-v-ax-kcl-k-l-aa-kcl-k-l-eh-v-el)	10
0.63	That's your headache.	(dh-ae-tcl-t-s-y-er-hv-eh-dx-ey-kcl-k)	13
0.6	Run-down, iron-poor.	(r-ah-n-dcl-d-aw-n-q-ay-er-n-pcl-p-ao-r)	12
0.49	Y'all wanna walk – walk, he said.	(y-ao-l-w-ao-n-ax-w-ao-kcl-pau-w-ao-kcl-k-iy-s-eh-dcl)	13
0.46	Pansies are gluttons.	(p-ae-n-z-iy-z-er-gcl-g-l-ah-tcl-en-d-z)	13
0.43	She seemed irritated.	(sh-iy-s-ey-m-dcl-d-ih-er-tcl-t-ey-dx-ix-dcl)	13
0.42	Where're you takin' me?	(w-er-y-ux-tcl-t-ey-kcl-k-ix-n-m-iy)	13
0.41	They're doin' it now.	(dh-eh-r-dcl-d-uw-ih-nx-ih-tcl-n-aw)	11
0.40	Yes, ma'am, it sure was.	(y-eh-s-epi-m-ae-m-ih-tcl-t-sh-er-w-ah-s)	13
0.40	Twenty-two or twenty-three.	(t-w-eh-n-tcl-t-iy-tcl-t-ux-ao-r-tcl-t-w-eh-n-tcl-t-iy-th-r-iy)	10
0.07	Boys and men go along the riverbank or to the alcoves in the top arcade.	(b-oy-z-ix-n-m-eh-n-gcl-g-ow-ax-l-ao-ng-n-ix-r-ih-v-er-bcl-b-ae-ng-kcl-k-q-ao-r-tcl-t-ux-dcl-d-iy-q-ae-l-kcl-k-ow-v-z-q-ix-n-dh-ix-tcl-t-aa-pcl-p-aa-r-kcl-k-ey-dcl-d)	34
0.07	But if she wasn't interested, she'd just go back to the same life she'd left.	(b-uh-dx-ih-f-sh-iy-w-ah-z-ix-n-ih-n-tcl-t-axr-s-tcl-t-ih-dcl-d-pau-sh-iy-dcl-jh-uh-s-gcl-g-ow-bcl-b-ae-kcl-t-ix-dh-ix-s-ey-m-l-ay-f-sh-iy-dcl-l-eh-f-tcl-t)	32
0.07	Why the hell didn't you come out when you saw them gang up on me?	(w-ay-dh-eh-hv-eh-l-dcl-d-ih-dcl-en-tcl-ch-ux-kcl-k-ah-m-aw-q-w-ix-n-y-ux-s-ao-dh-ix-m-gcl-g-ae-ng-ah-pcl-p-ao-n-m-iy)	31
0.06	You think somebody is going to stand up in the audience and make guilty faces?	(y-ux-th-ih-ng-kcl-k-s-ah-m-bcl-b-aa-dx-iy-ix-z-gcl-g-oy-ng-dcl-d-ix-s-tcl-t-ae-n-dcl-d-ah-pcl-p-ix-n-ah-q-aa-dx-iy-eh-n-tcl-s-eh-m-ey-kcl-g-ih-l-tcl-t-ix-f-ey-s-eh-z)	33
0.06	How much and how many profits could a majority take out of the losses of a few?	(hh-aw-m-ah-tcl-ch-ix-n-hv-aw-m-ax-nx-iy-pcl-p-r-aa-f-ax-tcl-s-kcl-k-uh-dx-ax-m-ax-dcl-jh-ao-axr-dx-iy-tcl-t-ey-kcl-k-ae-dx-ah-dh-ax-l-ao-s-ix-z-ax-v-ax-f-y-ux)	35
0.06	He may not rise to the heights, but he can get by, and eventually be retired.	(hh-iy-m-ey-n-aa-q-r-ay-z-tcl-t-ix-dh-ax-hv-ay-tcl-s-pau-b-ah-dx-iy-kcl-k-ix-ng-gcl-g-eh-q-bcl-b-ay-pau-q-ix-nx-iy-v-eh-n-ch-ix-l-iy-pau-b-iy-r-iy-tcl-t-ay-axr-dcl-d)	35
0.06	My sincere wish is that he continues to add to this record he sets here today.	(m-ay-s-en-s-ih-r-w-ih-sh-ix-z-dh-eh-tcl-hv-iy-kcl-k-ax-h-tcl-t-ih-n-y-ux-z-tcl-t-ax-h-q-ae-dcl-d-pau-t-ux-dh-ih-sh-r-eh-kcl-k-axr-dx-iy-s-eh-tcl-s-hh-ix-r-tcl-t-ax-h-dx-ey)	31
0.05	Then he fled, not waiting to see if she minded him or took notice of his cry.	(dh-ih-n-iy-f-l-eh-dcl-d-pau-n-aa-q-w-ey-dx-ih-ng-dcl-d-ix-s-iy-ih-f-sh-iy-m-ay-n-ix-dcl-d-hv-ih-m-pau-q-axr-tcl-t-uh-kcl-n-ow-dx-ih-s-ix-v-ix-z-kcl-k-r-ay)	32
0.01	We apply auditory modeling to computer speech recognition.	(w-iy-ax-pcl-p-l-ay-q-ao-dx-ix-tcl-t-ao-r-ix-m-aa-dx-el-ix-ng-tcl-t-uw-kcl-k-ax-m-pcl-p-y-ux-dx-er-s-pcl-p-iy-tcl-ch-epi-r-eh-kcl-k-ix-gcl-n-ih-sh-ix-n)	35