Towards Representative Subset Selection for Self-Supervised Speech Recognition

Abdul Hameed Azeemi 1  Ihsan Ayyub Qazi 1  Agha Ali Raza 1

Abstract

Self-supervised speech recognition models require considerable labeled training data for learning high-fidelity representations for Automatic Speech Recognition (ASR), which hinders their application to low-resource languages. We consider the task of identifying an optimal subset of training data to fine-tune self-supervised speech models for ASR. We make a surprising observation that active learning strategies for sampling harder-to-learn examples do not perform better than random subset selection for 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 WER in the early training epochs leads to better generalization performance. Extensive experiments on the wav2vec 2.0 model and TIMIT dataset show the effectiveness of COWERAGE, with up to 27% absolute WER improvement over active learning methods. We also report the connection between training WER and the phonemic cover and demonstrate that our algorithm ensures inclusion of phonemically diverse examples.

1. Introduction

There has been rapid progress in recent years towards improving self-supervised speech recognition models. Such models learn high-fidelity speech representations using a large amount of untranscribed data and use paired data for fine-tuning on the low-resource languages (Baevski et al., 2020; Hsu et al., 2021). However, a significant amount of labeled training data is still needed for the fine-tuning step, which presents a challenge when training these models for low-resource languages (Ren et al., 2019; Song et al., 2019). In addition, self-supervised learning approaches require large computational resources for obtaining a sufficiently rich representation for good performance on ASR and hence are not suited for compute-restricted environments (e.g., on-device computing). These constraints present opportunities for the construction of smaller yet representative subsets of data that can train ASR models using minimal resources.

The existing studies on data subset selection and active learning for speech and ASR systems (Wu et al., 2007; Yu et al., 2010; Hamanaka et al., 2010; Nallasamy et al., 2012; Wei et al., 2014; Fraga-Silva et al., 2015) focus heavily on the inclusion of phonetically rich text and word diversity to create balanced datasets. However, these methodologies are not useful for deep learning based ASR systems as the theoretical guarantees are not directly applicable due to the non-convex nature of self-supervised ASR models. 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 self-supervised ASR 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 self-supervised ASR without significantly sacrificing the test accuracy? What are the phoneme distributions of good subsets of training data? Can we analyze the training landscape of these ASR models 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 for low-resource languages and compute-restricted environments.
We find that in a standard speech dataset for training self-supervised ASR models, sampling only the hard-to-learn training examples on the basis of word error rate (WER) does not perform better than random pruning. This is in contrast to active learning 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 self-supervised ASR models, we propose COWERAGE, an algorithm we have designed for identifying training examples that are important for better generalization in self-supervised ASR models. We find that ensuring the coverage of examples on the basis of training WER in the early training epochs leads to better generalization performance 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 without explicitly learning any phoneme-level error model.

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 self-supervised ASR models.
- We present COWERAGE, an algorithm for selecting a subset of ASR training data that ensures uniform coverage of training WER via a bucketing approach.
- Our empirical evaluation on the TIMIT dataset (Garofolo et al., 1993) and wav2vec 2.0 (Baevski et al., 2020) show that training the model 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).
- 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, COWERAGE is able to select phonemically diverse examples, which results in a richer training subset.
- We compare the training trajectories of examples within the data subsets identified by the four pruning strategies. We find that COWERAGE also removes the harder to learn outliers within training examples, which increases the generalization accuracy.

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 \(\theta_l\) 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 pairs of audio and the corresponding sentence that was uttered. Our goal is to prune \(\mathcal{D}_l\) to obtain a subset \(\mathcal{B}_l\) such that the performance of self-supervised ASR model \(f(x;\theta)\) after fine-tuning on \(\mathcal{B}_l\) is better than random pruning. 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 (\(D\)).

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

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 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 harder-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 \(\mathcal{B}_l\) of the training data \(\mathcal{D}_l\) for fine-tuning a self-supervised speech model on ASR.

3.1. Strategy 1: Picking the hardest \(k\) examples

The first approach is to pick the top \(k\) training examples i.e., the ones with the highest WER (Algorithm 1). This replicates the active learning 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 \(p\).
Towards Representative Subset Selection for Self-Supervised Speech Recognition

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 $e$
$retainFraction \leftarrow 1 - p$
$retainSize \leftarrow retainFraction \times \text{len}(D_l)$
$W \leftarrow \text{sortDescending}(W)$
$B_t \leftarrow W[0 : retainSize]$

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 we consider it as one of the baselines to compare against when studying the training and test trajectories for the pruning strategies.

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 $e$
$retainFraction \leftarrow 1 - p$
$retainSize \leftarrow retainFraction \times \text{len}(D_l)$
$W \leftarrow \text{sortAscending}(W)$
$B_t \leftarrow W[0 : retainSize]$

3.3. Strategy 3: COWERAGE Subset Selection

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 $e$
$retainFraction \leftarrow 1 - p$
$B_t \leftarrow \text{EMPTYSET}$
$W \leftarrow \text{sortDescending}(W)$
$\text{buckets} \leftarrow \text{createBuckets}(W, \text{size} = b)$
for bucket in buckets do
    sampleSize $\leftarrow retainFraction \times \text{bucketSize}$
    $S \leftarrow \text{randomSample}(\text{bucket}, \text{sampleSize})$
    $B_t \leftarrow B_t \cup S$
end for

We now present another 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 verify later through multiple experiments (see Section 6).

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

With COWERAGE, we first compute the training WER for each example in $D_l$ and sort the examples in descending order of WER. Next, we construct multiple buckets of size $b$ and then randomly select $k$ training examples from each bucket, where $k$ is decided by the fraction of the dataset to be pruned e.g., for the pruning fraction of 0.3 and retaining fraction of 0.7, $k$ is 7. This ensures coverage of WER when selecting training examples. The overall algorithm is presented in Algorithm 3.

4. Configurations

4.1. Architecture

We use the wav2vec 2.0 (base) model (Baevski et al., 2020) for our experiments. wav2vec 2.0 is built with 12 transformer blocks having a model dimension of 768, an inner...
Towards Representative Subset Selection for Self-Supervised Speech Recognition

**Figure 2.** The subset of the TIMIT training data selected by each of the three strategies: bottom $k$ (left), top $k$ (middle) and COVERAGE (right). The pruning fraction is set to 0.5 and the WER selection epoch is 8.

**Table 1.** 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. The COVERAGE consistently demonstrates the lowest WER at various pruning fractions and selection epochs. WSE: WER Selection Epoch.

| WSE | Strategy | Pruning Fraction |
|-----|----------|------------------|
| 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 |
|     | COVERAGE | 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 |
|     | COVERAGE | 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 |
|     | COVERAGE | 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 |
|     | COVERAGE | 0.454 0.457 0.464 0.474 0.495 0.510 0.573 0.761 1.000 |

dimension of 3072 and eight attention heads. 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 wav2vec 2.0 which is pretrained on Librispeech 960h with the predictive coding objective. We fine-tune wav2vec 2.0 for ASR with a task-specific head using the Connectionist Temporal Classification (CTC) loss (Graves et al., 2006).

**4.2. Dataset**

We utilize the TIMIT Acoustic-Phonetic Continuous Speech Corpus (Garofolo et al., 1993) for fine-tuning wav2vec 2.0 on the downstream task of ASR. It is a phonetically rich corpus and comprises samples from 630 American English speakers. There are ten speech recordings by every speaker sampled at 16kHz. The train/test split is pre-defined to be 3.9h/1.4h. Additionally, there is a time-aligned segmentation of the words and phonemes, which makes it a very suitable dataset for empirically analyzing the impact of each example during training and testing.
Towards Representative Subset Selection for Self-Supervised Speech Recognition

Figure 3. The WER for test examples 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.

4.3. Baseline

We consider the baseline experiment of randomly pruning the train set of TIMIT corpus on multiple fractions and fine-tuning the ASR model on the generated subset. The performance evaluation is done through the WER on the test set of TIMIT (Fig. 1). The test WER on the complete training data is 0.392 whereas the character error rate (CER) is 0.140. The test WER on the complete dataset is higher as compared to the current benchmarks on LJSpeech and Librispeech since the pre-training and the fine-tuning data are from different sources (Librispeech and TIMIT, respectively) which mirrors the low-resource setting, where a single large corpora is not available.

5. Empirical Evaluation

We fine-tune the wav2vec 2.0 (base) model on the TIMIT 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. A bucket size of 10 is chosen for the COWERAGE strategy. The data subsets are then used to fine-tune the model for computing the final test accuracy. The subset selected by each of the three strategies is shown in Fig. 2.

We show the results of pruning experiments via different strategies across multiple selection epochs in Table 1. For each WER selection epoch and a subset selection strategy, we consider multiple fractions at which the dataset was pruned. 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 training epochs. At higher pruning fractions, the difference between the test WER for COWERAGE and the other pruning strategies increases e.g. at 16th epoch and 80% pruning, COWERAGE shows 27% absolute WER improvement over Top k strategy.

Table 2. 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 | 0.474 ± 0.005 |

In Fig. 3, we show the WER for the test examples varying with each epoch. We find that COWERAGE subset selection is able to achieve a significantly lower WER for the 25-50th percentile of examples as compared to the other two approaches.

5.1. Selecting within the buckets

The strategy proposed in the original approach 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 2 show that the random selection outperforms other strategies. We leave it for future work a more in-depth analysis of the different variations of the COWERAGE algorithm.

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 accuracy but no single offset size is consistently better than random pruning. An
Towards Representative Subset Selection for Self-Supervised Speech Recognition

5.3. Training Trajectories

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. 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 more usual examples instead of the very hard but rare examples. This is in agreement with the plots in Fig. 3 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.

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 relate with the training error during fine-tuning of the self-supervised speech recognition models. We now perform empirical analysis to verify claim 3.1.

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. 6). In the earlier training epochs, the examples with a relatively low (< 17) or a high (> 28) phonemic cover have a greater WER as compared to the examples with a moderate number of phonemes (17 ≤ phonemicCover ≤ 28). In the later epochs (≥ 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).

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 a lot of low WER sentences with a high phonemic cover (examples are presented in Appendix B). More importantly, this experiment validates our claim that ensuring the coverage of training WER in a particular subset leads to the inclusion of phonemically diverse training examples (Fig. 6).

Figure 6. The training WER and the phonemic cover of examples in TIMIT dataset compared over multiple training epochs. The phonemic cover is defined as the number of unique phonemes in a particular training example of the TIMIT dataset. 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 the particular epoch are computed by averaging over 10 runs.

Figure 7. The phonemic coverage ensured by the three pruning strategies for epoch 16 and 0.5 pruning fraction.

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 significant, with p-value < .01. The results are shown in Table 3.

Table 3. 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 (α = 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} * |

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; Paul 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 misclassified or ‘forgotton’ after being classified correctly during training. They find that rarely forgotten examples can be effectively eliminated from the training subset without affecting the generalization accuracy. Paul et al. (2021) study 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.

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 and active learning. 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.

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. As we also observed through the training trajectories, the COWERAGE algorithm is automatically removing some of the training outliers with the highest WER, which in turn is contributing towards better test accuracy.

The existing work on active learning and data pruning for ASR systems emphasize the importance of ensuring phonetically 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; Ni et al., 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. Another work (Mendonça et al., 2014) proposes a greedy algorithm for phonetically-rich triphone sentence selection. It works by recognizing the distance between the triphone distribution of the dataset and ideal uniform distribution and then creating a corpora with a more uniform distribution of triphones.

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. An interesting finding that relates to our study is that the phoneme-level error model for ASR personalization selects more challenging sentences (measured by WER) than random pruning but less challenging ones than the top 100 highest WER examples. This supports our finding about the downstream fine-tuning on ASR that ensuring the coverage of WER is a better selection strategy than just picking the hardest examples. 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 phoneme-level error model needs to be learned and 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. Our data pruning experiments were performed on a contrastive pre-trained self-supervised model (wav2vec 2.0). It would be worth exploring if the results are transferable to self-supervised approaches trained on a different objective such as masked prediction (e.g., HuBERT). Moreover the pre-training and fine-tuning data in our experiments were from different sources (Librispeech and TIMIT, respectively), thus mirroring the low-resource settings where a large-scale speech corpora is not available for pre-training as well as fine-tuning. While our approach is designed to be dataset agnostic, it remains to be empirically evaluated whether our methodology generalizes to conditions where the same data is used in the fine-tuning step. An interesting direction for future work is to extend subset selection to other ASR architectures and larger datasets where the gains due to data pruning may be further amplified for low-resource languages. 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 word error rate. A evaluation on wav2vec 2.0 and the TIMIT dataset show that COWERAGE performs better than random selection as well as active learning strategies that select harder-to-learn or easier-to-learn examples. We find that COWERAGE prefers excluding collective outliers and learning more common examples. Moreover, 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. We show that COWERAGE outperforms other subset selection strategies as it ensures phonemic diversity within the training examples.
Towards Representative Subset Selection for Self-Supervised Speech Recognition

References

Agarwal, C., D’souza, D., and Hooker, S. Estimating example difficulty using variance of gradients. *arXiv preprint arXiv:2008.11600*, 2020.

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

Awasthi, A., Kansal, A., Sarawagi, S., and Jyothi, P. 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.

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

Baldock, R. J., Maennel, H., and Neyshabur, B. Deep learning through the lens of example difficulty. *arXiv preprint arXiv:2106.09647*, 2021.

Coleman, C., Yeh, C., Mussmann, S., Mirzasoleiman, B., Bailis, P., Liang, P., Leskovec, J., and Zaharia, M. Selection via proxy: Efficient data selection for deep learning. *arXiv preprint arXiv:1906.11829*, 2019.

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

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

Fraga-Silva, T., Gauvain, J.-L., Lamel, L., Laurent, A., Le, V.-B., and Messaoudi, A. Active learning based data selection for limited resource stt and kws. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., and Pallett, D. S. DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc 1-1.1. NASA STI/Recon technical report n, 93:27403, 1993.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. 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.

Hamanaka, Y., Shinoda, K., Furui, S., Emori, T., and Koshinaka, T. Speech modeling based on committee-based active learning. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4350–4353. IEEE, 2010.

Han, J., Pei, J., and Kamber, M. *Data mining: concepts and techniques*. Elsevier, 2011.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *arXiv preprint arXiv:2106.07447*, 2021.

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

Jiang, S., Krauthgamer, R., Wu, X., et al. Coresets for clustering with missing values. *Advances in Neural Information Processing Systems*, 34, 2021.

Jubran, I., Sanches Shayda, E. E., Newman, I., and Feldman, D. Coresets for decision trees of signals. *Advances in Neural Information Processing Systems*, 34, 2021.

Karamcheti, S., Krishna, R., Fei-Fei, L., and Manning, C. D. Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. *arXiv preprint arXiv:2107.02331*, 2021.

Kaushal, V., Iyer, R., Kothawade, S., Mahadev, R., Doctor, K., and Ramakrishnan, G. 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.

Killamsetty, K., Sivasubramanian, D., Mirzasoleiman, B., Ramakrishnan, G., De, A., and Iyer, R. Grad-match: A gradient matching based data subset selection for efficient learning. *arXiv preprint arXiv:2103.00123*, 2021.

Kothawade, S., Beck, N., Killamsetty, K., and Iyer, R. Similar: Submodular information measures based active learning in realistic scenarios. *Advances in Neural Information Processing Systems*, 34, 2021.

Kothyari, M., Mekala, A. R., Iyer, R., Ramakrishnan, G., and Jyothi, P. Personalizing asr with limited data using targeted subset selection. *arXiv preprint arXiv:2110.04908*, 2021.

Lai, C.-I. J., Zhang, Y., Liu, A. H., Chang, S., Liao, Y.-L., Chuang, Y.-S., Qian, K., Khurana, S., Cox, D., and Glass, J. Parp: Prune, adjust and re-prune for self-supervised speech recognition. *arXiv preprint arXiv:2106.05933*, 2021.
Towards Representative Subset Selection for Self-Supervised Speech Recognition

Margatina, K., Vernikos, G., Barrault, L., and Aletras, N. 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.

Mendonça, G., Candeias, S., Perdigao, F., Toniaszio, R., Klautau, A., and Aluísio, S. A method for the extraction of phonetically-rich triphone sentences. In 2014 International Telecommunications Symposium (ITS), pp. 1–5. IEEE, 2014.

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

Nallasamy, U., Metze, F., and Schultz, T. Active learning for accent adaptation in automatic speech recognition. In 2012 IEEE Spoken Language Technology Workshop (SLT), pp. 360–365. IEEE, 2012.

Ni, C., Leung, C.-C., Wang, L., Chen, N. F., and Ma, B. 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.

Ni, C., Wang, L., Liu, H., Leung, C.-C., Lu, L., and Ma, B. 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.

Ni, C., Leung, C.-C., Wang, L., Liu, H., Rao, F., Lu, L., Chen, N. F., Ma, B., and Li, H. 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.

Paul, M., Ganguli, S., and Dz因果gai, G. K. Deep learning on a data diet: Finding important examples early in training. Advances in Neural Information Processing Systems, 34, 2021.

Raju, R. S., Daruwalla, K., and Lipasti, M. Accelerating deep learning with dynamic data pruning. arXiv preprint arXiv:2111.12621, 2021.

Ren, Y., Tan, X., Qin, T., Zhao, S., Zhao, Z., and Liu, T.-Y. Almost Unsupervised Text to Speech and Automatic Speech Recognition. In Proceedings of the 36th International Conference on Machine Learning, pp. 5410–5419. PMLR, 2019.

Saadatfar, H., Khosravi, S., Joloudari, J. H., Mosavi, A., and Shamshirband, S. A new k-nearest neighbors classifier for big data based on efficient data pruning. Mathematics, 8(2):286, 2020.

Schirrmeister, R. T., Liu, R., Hooker, S., and Ball, T. When less is more: Simplifying inputs aids neural network understanding. arXiv preprint arXiv:2201.05610, 2022.

Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. Mas: Masked sequence to sequence pre-training for language generation. In ICML, 2019.

Tolochinsky, E. and Feldman, D. Coresets for monotonic functions with applications to deep learning. CoRR, abs/1802.07382, 2018.

Toneva, M., Sordoni, A., Combes, R. T. d., Trischler, A., Bengio, Y., and Gordon, G. J. An empirical study of example forgetting during deep neural network learning. arXiv preprint arXiv:1812.05159, 2018.

Wei, K., Liu, Y., Kirchhoff, K., Bartels, C., and Bilmes, J. 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.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. Huggingface’s transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771, 2019.

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

Wu, Z., Zhao, D., Liang, Q., Yu, J., Gulati, A., and Pang, R. 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.

Yu, D., Varadarajan, B., Deng, L., and Acero, A. Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion. Computer Speech & Language, 24(3):433–444, 2010.

Zhen, K., Nguyen, H. D., Chang, F.-J., Mouchtaris, A., and Rastrow, A. 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. Implementation Details

A.1. Resources

We use a single 48GB NVIDIA A6000 GPU for running all the experiments.

A.2. Data

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.

A.3. Model

The HuggingFace (Wolf et al., 2019) implementation (Wolf et al., 2019) of wav2vec 2.0 (base) model is used which is based on the standard wav2vec2-base-960h fairseq implementation. The model is fine-tuned for ASR using the CTC loss. The data pruning strategies are implemented in Python, and the resulting subsets are used to fine-tune wav2vec 2.0 with CTC loss.

A.4. Training

In all experiments, wav2vec 2.0 (base) is fine-tuned with a batch size = 32, epochs = 20, learning rate 0.0004, weight decay = 0.005, warmup steps = 1000, 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 independent runs (with independent model initialization). The mean test WER is reported in Table 1 and the corresponding standard deviation is reported in Table 4.

| WSE | Strategy | 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 | ±0.004 | ±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 | ±0.003 | ±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 |

1https://catalog.ldc.upenn.edu/LDC93s1
2https://huggingface.co/facebook/wav2vec2-base-960h
3https://github.com/pytorch/fairseq/blob/main/examples/wav2vec/README.md
Table 5. Examples of short training sentences with a high training WER and long sentences with a low training WER. The training WER is calculated by averaging 10 runs.

| Av. Training WER | Text | Phonemes | Phonemic Cover |
|------------------|------|----------|----------------|
| 0.63             | Twelve o’clock level. | (t-w-eh-l-v-ax-k-cl-k-l-aa-k-cl-k-l-eh-v-cl) | 10 |
| 0.63             | That’s your headache. | (dh-ae-tcl-t-s-y-er-hv-eh-dx-ey-k-cl-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-aq-k-cl-pau-w-ao-k-cl-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- pcl-dl-ih-er-tcl-t-ey-dx-ix-dcl) | 13 |
| 0.42             | Where’re you takin’ me? | (w-er-y-ux-tcl-t-ey-k-cl-k-ix-n-miy) | 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-riy) | 10 |
| 0.07             | Boys and men go along the riverbank or to the alcoves in the top arcade. | (b-o-y-z-ix-n-m-eh-n-gcl-g-ow-ax-l-ao-ng-n-ixr-ih-v-er-bcl-b-ae-ng-k-cl-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-ixn-ih-n-tcl-t-axr-s-tcl-t-hih-dcl-d-pau-sh-iy-dcl-jh-uh-uh-s-gcl-g-ow-bcl-b-ae-kcl-t-ix-dh-ix-s-e-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-ahm-maw-q-w-ix-n-y-ux-s-ao-dh-ix-m-gcl-g-ae-ng-ah-pcl-p-ao-n-miy) | 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-o-y-ng-dcl-d-ix-s-tcl-t-ae-n-dcl-d-ah-pcl-p-ix-n-ah-q-aa-dx-dy-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-aa-ax-xf-clkl-k-k-l-cl-k-d-uh-ux-dx-ax-m-ax-dcl-jh-ao-axr-dx-iy-tcl-t-ey-kcl-k-ae-dx-ab-dh-ax-l-ao-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-ih-m-ey-n-aa-q-r-ay-z-tcl-i-ix-dh-ax-hv-ay-tcl-s-pau-b-ah-dx-iy-kcl-k-ix-ng-gcl-g-eh-q-cll-b-ay-pauq-ix-ix-n-y-iy-v-eh-n-ch-ix-l-iy-pau-b-iy-ry-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-s-en-s-ih-r-w-ih-sh-ix-z-dh-eh-tcl-hv-iy-kcl-k-ax-h-tcl-t-iy-n-ux-z-tcl-t-ax-h-q-ae-dcl-d-pau-t-ux-dh-ih-sh-er-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-leh-dcl-d-pau-n-aa-q-w-ey-dx-ih-ng-dcl-d-ix-siy-iy-f-leh-may-n-ay-dcl-d-hv-ihm-pau-paq-axr-tcl-t-uh-kcln-ow-dx-ih-s-ix-v-x-z-kcl-k-ry) | 32 |
| 0.05             | Then he fled, not waiting to see if she minded him or took notice of his cry. | (dh-ih-n-iy-f-leh-dcl-d-pau-n-aa-q-w-ey-dx-ih-ng-dcl-d-ix-siy-iy-f-leh-may-n-ay-dcl-d-hv-ihm-pau-paq-axr-tcl-t-uh-kcln-ow-dx-ih-s-ix-v-x-z-kcl-k-ry) | 32 |
| 0.01             | We apply auditory modeling to computer speech recognition. | (w-y-ix-pcl-p-l-ay-q-ao-dx-ix-tcl-t-ar-x-m-aa-dx-ixl-ixng-tcl-t-uh-kcl-k-k-m-pcl-p-y-ux-dx-er-s-pcl-p-iy-tcl-ch-epi-eh-kcl-k-ix-gcl-n-ih-sh-ix-n) | 35 |