

Ask2Mask: Guided Data Selection for Masked Speech Modeling

Murali Karthick Baskar¹, Andrew Rosenberg, Bhuvana Ramabhadran², *Fellow, IEEE*, Yu Zhang³,
and Pedro Moreno, *Member, IEEE*

Abstract—Masked speech modeling (MSM) methods such as wav2vec2 or w2v-BERT learn representations over speech frames which are randomly masked within an utterance. While these methods improve performance of Automatic Speech Recognition (ASR) systems, they have one major limitation. They treat all unsupervised speech samples with equal weight, which hinders learning as not all samples have relevant information to learn meaningful representations. In this work, we address this limitation. We propose ask2mask (ATM), a novel approach to focus on specific samples during MSM pre-training. ATM employs an external ASR model or *scorer* to weight unsupervised input samples in two different ways: 1) A fine-grained data selection is performed by masking over the highly confident input frames as chosen by the scorer. This allows the model to learn meaningful representations. 2) ATM is further extended to focus at utterance-level by weighting the final MSM loss with the utterance-level confidence score. We conduct fine-tuning experiments on two well-benchmarked corpora: LibriSpeech (matching the pre-training data) and Commonvoice, TED-LIUM, AMI and CHiME-6 (not matching the pre-training data). The results substantiate the efficacy of ATM on significantly improving the recognition performance under mismatched conditions (up to 11.6% relative over published results and upto 4.46% relative over our internal baseline) while still yielding modest improvements under matched conditions.

Index Terms—Self-supervision, Wav2vec2, Data selection, Domain mismatch.

I. INTRODUCTION

SELF-TRAINING and self-supervised training techniques rely on huge amounts of unlabeled speech or text data for better generalization. The self-training techniques such as pseudo-labeling [1], [2] and student-teacher training [3] have shown promising improvements by incorporating the data selection process. This data selection step removes pseudo-labels with less confidence as denoted by the teacher model before feeding the input to a student model. [4] shows that self-training and self-supervised training are complementary to each other and

also show that self-supervised models act as good initialization for self-training techniques. Self-supervised training [5] is a representation learning approach which implicitly learns patterns in the data without relying on explicit labels. Masked speech modeling (MSM) is the recent and successful self-supervised learning technique, thanks to the advent of BERT [6] in NLP which inspired learning speech representations from masked inputs. MSM techniques such as wav2vec2 [7], HuBERT [8] and w2v-BERT [9] have shown considerable gains across various down-stream speech tasks and have become the go-to models for ASR.

Unfortunately, MSM does not have a data selection scheme to discard the irrelevant input samples and instead imposes burden on the training criterion to learn the relevance of the input samples in learning meaningful representations. [10] noticed the impact of not selecting relevant data from the huge amounts of unsupervised data during pre-training by showing degradation in ASR performance when fine-tuned to a target dataset with limited data. To mitigate this constraint, [11] introduced substantially more fine-tuning data related to the target dataset but did not achieve satisfactory results. [10] attempted to solve this issue by heuristically selecting the data from a closed set of unsupervised speech databases or by pooling in data relevant to target dataset along with the existing pre-training dataset. However, this data selection approach is not done within the existing pre-training dataset and it is not completely empirically motivated.

In this study, in order to break the above limitation of the MSM techniques, we propose a simple strategy named *ask2mask* (ATM) to incorporate data selection within a chosen pretraining dataset.

- In ATM, the masking is done over the input samples or speech frames with higher confidence as determined by the scorer. This is contrary to the random selection of frames to be masked in conventional MSM models. We hypothesize that this guided selection of frames to be masked allows the model to focus on the frames which can provide meaningful representations. The scoring model used in this work is necessarily a speech recognition model trained on small amount of data and provides frame-level confidence for each input.
- The ATM technique is further extended to exploit the confidence values provided by the scorer by directly using them to re-weight the MSM loss. We denote this approach as ATM with loss scaling (ATM+S). It allows the model

Manuscript received 15 January 2022; revised 17 June 2022; accepted 21 June 2022. Date of publication 24 June 2022; date of current version 14 October 2022. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Abdelrahman Mohamed. (*Corresponding author: Murali Karthick Baskar.*)

Murali Karthick Baskar is with the Faculty of Information Technology, Brno University of Technology, 601 90 Brno-stred, Czechia (e-mail: baskar@vutbr.cz).

Andrew Rosenberg, Bhuvana Ramabhadran, Yu Zhang, and Pedro Moreno are with the Google, Inc., Mountain View, CA 94043 USA (e-mail: rosenberg@google.com; bhuv@google.com; ngyuzh@google.com; pedro@google.com).

Digital Object Identifier 10.1109/JSTSP.2022.3186162

training to focus on certain *utterances* by down scaling the utterances with low-confidences.

Similar to our work based on masking with external guidance, there is work in NLP that also benefit by incorporating masking with knowledge. In [12], masking is done at phrase-level segments in BERT and has shown to learn semantic dependencies. In [13], phonetic knowledge is injected to mask over phonetic segments to perform spectral augmentation. Phonetically motivated masking scheme is proposed in [14] to improve multiple downstream speech tasks. PEGASUS [15] model masks the input text based on their ROUGE score provides better self-supervised representations for text summarization and is more closer to the idea behind our work.

Our ATM approach is primarily motivated based on the recent work by [16] on semi-supervised learning of conventional ASR systems which shows that performing data selection at frame-level or token-level on unsupervised data provides better performance. The importance of pruning out the input samples at frame-level has been studied in [17] to improve both classification and regression tasks. Few works on unsupervised learning also highlight the importance of weighting the data based on its confidence [18]–[20]. We hypothesize that ATM can leverage the effect of data selection within a particular training corpus to further enhance the recognition performance of MSM techniques.

To summarize, our contributions are listed as follows:

- *Novelty*: To the extent of our knowledge, ATM is the first approach to incorporate a within-corpus data selection strategy in MSM. We also show that data selection can be simply performed inside MSM by guided selection of frames to be masked using a scorer model.
- *Technical contributions*: We provide two simple strategies to incorporate data selection into MSM pretraining by applying the confidence of the scorer: 1) choosing the data at frame-level by applying guided masking 2) soft weighting the data at utterance level by scaling the MSM loss of each utterance with its corresponding confidence score. ATM is designed to be compatible to all MSM based pre-training techniques.
- *Empirical study*: Analysis is done to find an optimal masking percentage for ATM and we highlight the effectiveness of ATM across varying masking percentages. The importance of masking frames with high confidence is substantiated by empirically comparing it with masking low confident frames and random frames respectively. Experiments are performed on AMI data which is from a distinct condition compared to Libri-light corpus used for MSM based pretraining. The results confirm the importance of ATM by improving the recognition performance on evaluation sets of AMI by a significant margin.

II. MASKED SPEECH MODELS (MSM)

In this section, we formally define the masked speech modeling (MSM) technique and brief primary instantiations including wav2vec2 and w2v-BERT. The technique can be formulated by defining input speech sequence $\mathbf{X} = [x_1, x_2, \dots, x_{T'}]$, where

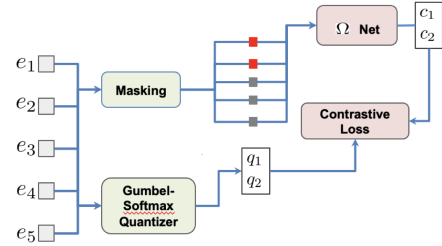


Fig. 1. Working procedure of Wav2vec2-conformer model as described in section II-C. The encoded representations are masked and passed to context network Ω and the resulting output c_j is learnt to be closer to quantized output.

x_t is the log Mel-filterbank feature vector at time t . \mathbf{X} is sent to the feature encoder Φ to obtain the encoded representations $\mathbf{E} = \Phi(\mathbf{X})$. The feature encoder contains convolutional layers performing subsampling at a factor of 4 and reducing the total number of frames of an utterance from T' to T to get $\mathbf{E} = [e_1, e_2, \dots, e_T]$. \mathbf{E} is then sent to two parallel modules: 1) masking component, and 2) quantizer.

A. Masking

The idea behind masking input samples and predicting them was initially proposed in BERT [6] and later adopted to speech [7] with modifications to suit the characteristics of speech input. The masking is done over sets of frames or blocks b_1, b_2, \dots, b_K and accommodates overlap between blocks. Here K is the number of masked blocks in a randomly masked encoded sequence $\tilde{\mathbf{E}}$. The importance of block masking is motivated by the improvements observed in Span-BERT by [21] and ERNIE [12]. The block $b_k = [i_k, c]$, where i_k is the starting index of the masked block and c is the corresponding right context size denoting the number of consecutive speech frames. Here i_k are randomly sampled from a uniform distribution. It has been empirically observed by [7] that 49% of the frames are masked and $c = 10$ is chosen as the golden ratio to attain best representation during pre-training.

B. Quantizer

Gumbel-softmax quantizer component Ψ is used to get quantized representations which act as targets for wav2vec2 and w2v-BERT models. These quantized representations align to phonetic units as described in [7]. Each quantized vector is of L dimensions which denote the number of targets or codes used in a codebook. Each incoming input \mathbf{E} is projected to L dimensions within the quantizer before applying the Gumbel-softmax.

C. Context Network and MSM Loss

Wav2vec2-conformer (w2v2-cfr): In this model type, as in Fig.1, the unmasked sequence \mathbf{E} is sent to Ψ to get $\mathbf{Q} = \Psi(\mathbf{E})$, where $\mathbf{Q} = [q_1, q_2, \dots, q_T]$ as described in [7].

The masked sequence $\tilde{\mathbf{E}}$ is fed to the context network Ω which contains conformer blocks to learn contextual representations from the input. $\mathbf{C} = \Omega(\tilde{\mathbf{E}})$ denotes the output of the context network. The contrastive loss $\mathcal{L}_{ctr}(c_j, q_j)$ objective is computed

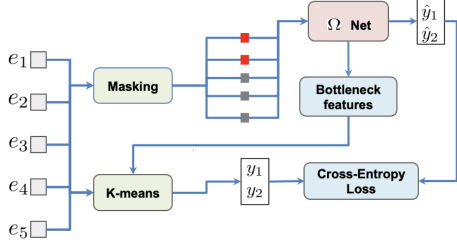


Fig. 2. Working procedure of HuBERT-conformer model as described in section II-C. The k-means cluster ids act as labels and they are refined using the bottleneck features extracted from the context network itself.

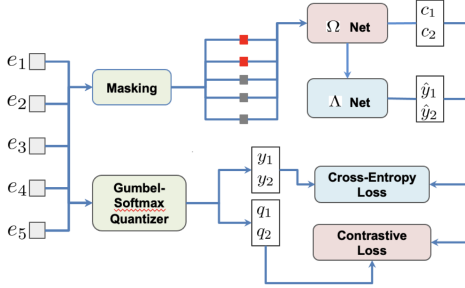


Fig. 3. Working procedure of W2V-BERT model as described in section II-C. Cross-entropy loss is computed between predictions of context network Λ and the quantized labels y_j . Contrastive loss is computed in parallel as in wav2vec2.

between the quantized representation q_j and context network output $c_j \in \mathbf{C}$ for all masked time instances $j \in J$. Diversity loss \mathcal{L}_{div} is computed as an auxiliary objective in wav2vec2 to force the model to choose diverse codes in the quantization codebook. Detailed description of \mathcal{L}_{div} is in [7]. The final training objective is denoted as:

$$\mathcal{L}_{wv} = \mathcal{L}_{ctr} + 0.1 \cdot \mathcal{L}_{div}, \quad (1)$$

where $\mathcal{L}_{ctr} = \sum_{j=1}^J \mathcal{L}_{ctr}(c_j, q_j)$.

HuBERT-conformer: Figure 2 is another variant of wav2vec2-conformer model with two major differences: 1) Targets are k-means cluster ids which are computed over a small portion of input 2) Cross-entropy loss $\mathcal{L}_{ce}(\hat{y}_j, y_j)$ is computed between the prediction of the context network \hat{y}_j and the k-means cluster id target y_j .

W2V-BERT: Figure 3 This model marries the concept of wav2vec2 and BERT model by including an additional context network Λ containing conformer blocks in addition to Ω as in wav2vec2. The Λ receives the output of the Ω and strives to further learn refined contextual information to get $\mathbf{H} = \Lambda(\mathbf{C})$. The targets of w2v-BERT y_j is computed by taking an argmax over the codebook dimensions L of quantized representations $q_{j,l}$ as:

$$y_j = \arg \max_l q_{j,l}, l \in L \quad (2)$$

Finally, the cross-entropy loss $\mathcal{L}_{ce}(\hat{y}_j, y_j)$ is computed between the prediction $\hat{y}_j = \text{softmax}(h_j)$ and the target y_j over the masked time instances J . The final training objective $\mathcal{L}_{wb} =$

$\mathcal{L}_{ce} + \mathcal{L}_{wv}$ is a combination of cross-entropy loss and wav2vec2 loss.

III. ASK2MASK (ATM)

The primary reason to employ pre-training models is to exploit the abundantly available unsupervised data for improving ASR under limited availability of supervised data. While the MSM models such as wav2vec2 and w2v-BERT described in Section II exploit the unsupervised data, they treat each data with equal weight for computing the final objective. Instead, we generate a score s_t for each encoded frame e_t . This is used to select relevant data in a fine-grained manner during masking for computing the loss objective. Here, we hypothesize that pre-training with data that closely resembles the target domain leads to better recognition performance after fine-tuning.

A. Methodology

For each encoded feature frame $e_t \in \mathbf{E}$, the scorer emits probabilities $p(v_t = l | \mathbf{E})$; $l \in \mathbf{L}$ of the frame belonging to a particular label. The scorer model is a CTC based frame-synchronous ASR model separately trained with a limited amount of data. Our initial intuition was to chose the scorer's training data to match the target data condition, however our empirical analysis in (cf. Section VI-C) shows that the performance is agnostic to the scorer model's training data. Finally, the confidence score s_t of the frame is defined as the maximum probability across all labels:

$$s_t = \max_l p(v_t = l | \mathbf{E}). \quad (3)$$

We sample K masking start indices $\{i_1, \dots, i_k\}$ with probabilities:

$$p(i_k = t) = \frac{s_t}{\sum_{v \notin \{i_1, \dots, i_{k-1}\}} s_v} \cdot \delta_{t \notin \{i_1, \dots, i_{k-1}\}}, \quad (4)$$

That is, we sample beginning frames with probability proportional to the scores of each frame. The indicator function $\delta_{t \notin \{i_1, \dots, i_{k-1}\}}$ ensures that we sample *without* replacement. This is the key difference between ATM and the random masking in prior works as described in Section II-A. Prior works uniformly sample the start indices of each masking block $b_{1:K}$, while the ATM uses the probability distribution induced by the scorer. K is determined by the percentage of frames to be masked. In ATM, we choose to also include the neighboring frames in the right from each selected frame.

We hypothesize that frames with maximum confidence from an external scoring model will be 1) easiest to learn using an MSM training criteria and 2) most informative in for pretraining to facilitate fine-tuning. Conversely, the lowest confidence frames, those more confusable to an external scoring model, will be the least reliably learned by MSM and least informative for pretraining.

The resulting frames are sent as input to the MSM architecture and the final loss objective \mathcal{L} is determined by either of the MSM objectives \mathcal{L}_{wv} or \mathcal{L}_{wb} described in Sections II-C. This modified

training objective allows the model to focus on learning from gradients calculated from the frames with high confidences.

B. ATM With MSM Loss Scaling (ATM+S)

The ATM loss is computed over the frames with high confidence performing a fine-grained data selection within a u^{th} speech sequence \mathbf{X}_u . Utterances with higher average frame confidence as measured by the scorer are accorded higher value than those with more confused frames. To perform data selection at a coarser utterance level, confidence scores s_u are computed as:

$$s_u = \frac{1}{T} \sum_{t=1}^T s_{t,u}. \quad (5)$$

For simplicity, we denote $s = s_u$ and the MSM loss computed over each masked frame is scaled by s to impose the importance of a particular utterance u . The final training objective \mathcal{L}'_{atm} of a particular speech sequence is denoted as:

$$\mathcal{L}'_{atm} = s \cdot \mathcal{L}_{atm}. \quad (6)$$

C. Probability as Confidence Measure in ATM

The ATM uses probability as a simple form of confidence measure to each frame. The analysis of confidence measures for semi-supervision in ASR has been done in [16] and they show that posterior probability acts as a reliable confidence measure for frame, word and sentence based data selection. They also perform an extensive analysis on using the posteriors for hybrid ASR systems. Based on the motivation from this work we chose to use softmax probability directly as our confidence score. A similar observation has been noted in [17], where the usage of softmax probability directly as a confidence measure has been applied to select relevant data samples during training. We also experimented with Entropy and exponential scaling or log scaling on softmax probabilities as confidence measure, but it did not fetch advantage over simple usage of probability.

IV. RELATED WORKS

Masking input has been widely explored to improve the ASR performance. Random masking is a simple yet effective strategy and has shown benefits across MSM models such as wav2vec2, HuBERT and w2v-BERT.

ATM manipulates the masking strategy in MSM to perform data selection on unsupervised pretraining data. However, this is not applicable to all pseudo-label based data selection. Several works have been performed to smartly perform filtering of unsupervised data where the pseudo-label acts as targets and is trained using supervised objectives. Unsupervised domain adaptation via uncertainty driven self-training [22] extracts hypotheses (H_d) from multiple ASR models by enabling dropout layers and changing the random seed. A reference hypothesis (H_r) is compared with H_d and if the maximum of all the scores is above a threshold, the reference hypothesis is selected as the pseudo-label and the corresponding utterance participates in training. In MSM, the pseudo-label sequences are not used and the targets are simple quantized representations. Another

main difference between self-training and MSM is that they don't involve decoder during pretraining which restricts the use of sequence as targets. Moreover, the MSM training and self-training are complementary as in [bigSSL] and ask2mask based MSM can serve as a better initialization for the encoder model used in self-training.

ATM involves a deterministic masking scheme and target domain focused MSM pretraining. REALM [23] is a MLM based target domain focused pretraining. A knowledge retriever and knowledge augmented encoder is used in REALM to retrieve the target knowledge representation and directly integrate in during pretraining and finetuning. ATM is a simplified approach and doesn't include scorer during finetuning unlike REALM where the knowledge network and retriever is used during finetuning and hence doesn't increase model complexity. PMI Masking [24] is another work which shows the importance of deterministic masking as in ask2mask. Here PMI selects n-grams based on their PMI score and masks them during MLM pretraining. The motivation of PMI is to learn whole words and ask2mask uses confidence scores from probability distribution of the scorer to select indomain samples from the huge unsupervised data.

V. EXPERIMENTAL SETUP

All experiments including pre-training and fine-tuning are performed using 80 dimensional log Mel-filterbank features computed over the sampled 16 kHz audio. Datasets (such as AMI) contains wideband audio and are downsampled to 16 kHz. We evaluate with the test-other (LibriSpeech partition) to show the importance of ATM on matched data conditions, while IHM-eval and SDM-eval (AMI partitions) is used to validate the model under mismatched conditions.

A. Datasets Used

Pretraining (PT): Libri-light (LL-60 k) dataset contains 60 k hours of unlabeled speech and is used to pre-train all MSM models. LL-60 k is the most widely used large unsupervised speech corpus for various PT techniques. Each input speech sequence is constructed by first randomly selecting 32-64 seconds segments from the original utterance. From these segments, a contiguous 32 s region is extracted from a random starting point on-the-fly during MSM PT as described in [25].

Finetuning (FT): Different target datasets including 1) 100 & 960 hours of Librispeech (LS-100 & LS-960) [26]. 2) 100 hours of AMI and 3) speechstew (approx. 5 k hours) [11] are used to perform our FT experiments. Each dataset used is specific to a certain target data condition, for instance LS-960 is closely matches the LL-60 k, AMI dataset is distinct from the LL-60 k condition and it contains speech from two kinds of microphones (i) Independent head microphone (IHM). (ii) single distant microphone (SDM). SpeechStew is composed of datasets chosen from multiple conditions to create a mixed domain aggregate corpus. Processing details are described in [11].

Evaluation: We hypothesize that evaluation over AMI using IHM-eval and SDM-eval reveals the effectiveness of ATM in providing informative samples for better representation learning.

We also evaluate using evaluation sets from Tedlium and Common voice as their training counter parts are used in SpeechStew based FT. Finally, we also evaluate using CHiME-6 [27] without using any FT data from CHiME-6 training set to compare the performance of ATM on completely unseen target dataset.

Scorer training data: A CTC [28] based conformer model with 100 M parameters is trained on LS-100 (“LS-scorer”). A similar model is also trained on AMI (“AMI-scorer”). Word-piece model (WPM) with 1024 tokens are used as labels for training the scorer models. All the results in this paper use “LS-scorer” besides the comparison Section VI-C.

B. MSM Architecture

W2v2-cfr: This is a wav2vec2 with conformer based context network which first encodes the filterbank features using two 2D convolutional layers with strides (2,2). Model has 100 M/600 M parameters and is denoted as “w2v2-cfr-L/XL”. HuBERT-cfr-L/XL is similar to w2v2-cfr-L/XL - it differs in using the k-means based quantizer with 1024 targets and computes the cross-entropy loss as described in [8]. The “L/XL” size models contains context network Ω 12/24 conformer layers with 8 attention heads and 1024 hidden dimensions.

W2v-BERT: W2v-BERT is explored using two model sizes: one with 100 M parameters denoted as “w2v-BERT-L” and containing 2 conformer layers in context net Ω and 4 conformer layers in Λ . A 600 M parameter model is denoted as “w2v-BERT-XL” contains 8 conformer layers in Ω and 24 conformer layers in Λ . Each conformer block contains 1024 hidden dimensions with 8 attention heads, kernel size of 5 with local context of 128. The remaining architecture is identical to the configuration defined in [9].

C. PT and FT Configuration

The models L/XL are trained with a global batch size of 512/2048 on 64/256 Google TPU V3 cores for 2-4 days respectively. Adam optimizer is used with a learning rate schedule (Section 5.3 of [29]) with $2e-3$ as peak learning rate and 25 k warmup steps. The model training configuration follows similar procedure as described in [25].

The FT is done by employing the context network from the PT model by adding a final layer with 1024 WPM units learnt using the RNN-T objective function [30]. The FT is done on w2v-BERT-XL, w2v2-cfr-XL and HuBERT-cfr-XL after 400 k PT model updates. The w2v-BERT-L model is FT after 900 k PT model updates. w2v-BERT-L is used to initially perform wide range of analysis and hyper-parameter optimization on ATM. w2v-BERT-XL is finally used to compare the results of ATM across existing works in literature. w2v2-cfr-XL and HuBERT-cfr-XL are also used in our experiments. All these models are trained with the same configuration as in [25].

VI. ATM ANALYSIS

The empirical study on ATM is done primarily using w2v-BERT-L since this generates the best WER performance across similarly sized models (cf. Fig. 5). The pre-trained models are

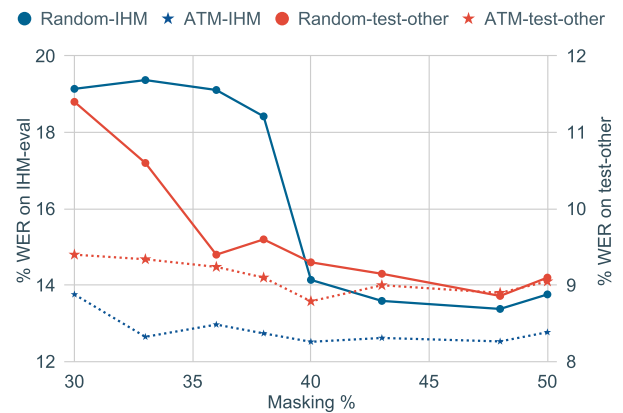


Fig. 4. Recognition performance of w2v-BERT with ATM and random masking on IHM-eval and test-other sets by varying the masking percentage during pre-training. The FT is performed on LS-100 for evaluating test-other, while IHM-eval is evaluated with model FT with AMI. Random masking shows a substantial shift in performance when varying the masking from 30% to 40%, while the ATM remains robust to changes in masking percentage.

fine-tuned with either LS-100 or AMI. The resulting finetuned models are evaluated on IHM and SDM evaluation sets to understand the domain generalization aspect of ATM. Librispeech evaluation sets are used in unison to study how ATM behaves under matching domain condition. These experiments are performed with using the loss scaling (it will be discussed in Section VI-F).

A. Masking Percentages

The number of masked frames within an utterance plays a key role in masked input learning and in this study, we vary the masking percentages from 30% to 50% to determine the best percentage for ATM approach. Previous works on wav2vec2 ([7]) showed that masking 49% of the frames is ideal for 30 s utterance and this has been followed subsequent works such as HuBERT and w2v-BERT. In case of ATM, this can differ as the frames selected are of higher confidences. Fig. 4 shows that ATM achieves its “sweet spot” with 40% masking for both IHM-eval and test-other set. Interestingly ATM’s performance is stable across large variations in masking rates with relatively good performance with masking rate as low as 30%. This is a significant difference from the uniform sampling of prior work which suffers significant drop in performance as the masking rate goes below 40%. The result indicates that masking the right set of frames, which ATM aims to do, is able to promote more stable performance. For instance, ATM achieves a %WER of 12.65 with 33% masking and 12.52 with 40% masking on IHM-eval respectively as shown in Fig. 4. The recognition performance on test-other and IHM-eval improves over baseline from 8.86% to 8.79% and 13.38% to 12.52% respectively by using ATM.

B. ATM Masking Strategies

The default setting of ATM is chosen based on a hypothesis that those frames that are scored with high-confidence from an external scoring model will be most useful as candidates

TABLE I

PERFORMANCE COMPARISON (IN %WER) ON AMI EVALUATION SETS USING w2-BERT WITH RANDOM MASKING (BASELINE) AND WITH ATM USING HIGH, LOW AND MIXED CONFIDENCE SCORES FROM THE SCORER. THE FT IS DONE WITH AMI.

Model	Confidence Level	PT-LL, FT-AMI	
		IHM-eval	SDM-eval
Baseline	Random	13.38	31.63
	High	12.52	27.34
	Low	14.14	37.77
ATM	Mixed	13.96	30.51

for MSM pretraining. This hypothesis is interrogated in this section by analysing the impact of choosing the frames with low confidences or equal mix of both high and low confidence frames (Mixed). For masking low confident frames, we modify the score in (3) as:

$$s_t = 1 - \max_l p(v_t = l | \mathbf{E}). \quad (7)$$

We evaluated these three masking strategies of ATM on both IHM and SDM evaluation sets. Table I shows the comparison between these sampling strategies. We observe that masking high confident frames are consistently better than masking the low confidence counterparts. In fact, “Low” confident frames perform worse than the baseline with random masking. Finally, we observe that performance of “Mixed” falls between that of “High” and “Low”. The “Mixed” strategy is similar to random masking, as both high and low confidence frames are selected. This similarity is also reflected in comparable performance between “Mixed” and random masking. These results provide support for our initial hypothesis that masking frames with high confidence leads to better pre-training.

C. How to Choose the Scoring Model

The scorer used in this work is a speech recognition model (100 M parameters) trained in a supervised fashion. The scorer is chosen based on the target downstream task and in addition to this, the scorer needs to be frame-synchronous to provide confidence for every frame in a speech sequence. In this work, we use a frame-synchronous ASR system as the scorer by employing the connectionist temporal classification (CTC) objective. The CTC is preferred over the RNN-T by analysing the reliability of the frame-level predictions. To analyse the importance of the supervised data used to train the scorer, we trained two scorer models: LS-scorer and AMI-scorer are CTC models trained with LS-100 and AMI dataset respectively. The AMI-scorer outperforms on SDM-eval by improving the %WER from 27.34 to 27.00. Surprisingly, our results on Table II, shows that the results on IHM-eval using an LS-scorer are comparable to the AMI-scorer. Evaluation on test and test-other shows that LS-scorer is better than AMI-scorer on both sets. Based on these observations, we choose the LS-scorer as the universal scoring model for all ATM based pre-trained models regardless of the target domain (eg: AMI) used in our experiments. Table II shows that although matching the scorer to the target domain improves the performance, the difference is not significant.

TABLE II

CROSS ANALYSIS OF ATM PERFORMANCE (IN %WER) USING AMI AND LS SCORERS. THE FT IS DONE ON LS-100 TO EVALUATE THE TEST AND TEST-OTHER, WHILE THE FT IS DONE ON AMI TO EVALUATE USING IHM-EVAL AND SDM-EVAL

Evalset	LS-scorer		AMI-scorer	
	IHM-eval	SDM-eval	IHM-eval	SDM-eval
test	3.89	3.93	3.89	3.93
test-other	8.92	9.68	8.92	9.68
IHM-eval	12.52	12.3	12.52	12.3
SDM-eval	27.34	27.00	27.34	27.00

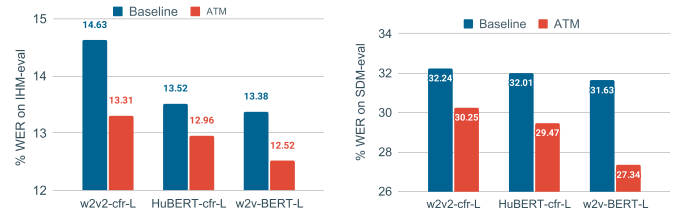


Fig. 5. Performance comparison of different MSM architectures with and without applying ATM on IHM-eval and SDM-eval in AMI. All these models are FT using AMI. Here “cfr” refers to conformer.

TABLE III

PERFORMANCE COMPARISON OF DIFFERENT MSM ARCHITECTURES WITH AND WITHOUT APPLYING ATM ON ALL EVALUATION SETS ON LIBRISPEECH.

Model	Type	PT-LL, FT-LS100			
		dev	dev-other	test	test-other
w2v-BERT-L	Baseline	3.78	8.86	3.85	9.32
	ATM	3.71	8.97	3.89	8.92
w2v2-cfr-XL	Baseline	2.5	4.7	2.6	4.9
	ATM	2.4	4.6	2.5	5.0
HuBERT-cfr-XL	Baseline	2.5	4.7	2.6	5.0
	ATM	2.5	4.6	2.5	5.0
w2v-BERT-XL	Baseline	2.4	4.4	2.5	4.6
	ATM	2.3	4.4	2.4	4.7

D. Consistency Across Different Architectures

Fig. 5 shows that ATM consistently outperforms on both IHM-eval and SDM-eval across multiple MSM architectures including wav2vec2 and HuBERT. In the case of IHM-eval, ATM attains a relative improvement of 9% over w2v2-cfr-L, 4% relative improvement over HuBERT-cfr-L and 5% relative gain over w2v-BERT-L baseline models respectively. W2v2-cfr-L using ATM obtained 6.2% relative improvement over its baseline counterpart and HuBERT-cfr-L with ATM attained 7.9% rel. improvement over HuBERT-L baseline on SDM-eval respectively. On the other hand w2v-BERT-L baseline is better compared to w2v2-cfr-L and HuBERT-cfr-L on both IHM-eval and SDM-eval by achieving 12.52% and 27.34% WER respectively.

E. Analysis of ATM on Librispeech

Experimental analysis is conducted using different model architectures to validate the effect ATM on LS-100 and are present in Table III. To study the impact of increasing the model parameters from “L” size to “XL” size, we FT on LS-100 using MSM models with XL size and the results are in Table III. We did not find any consistency in the performance across the

TABLE IV
%WER OF ATM+S BY FINE-TUNING ON AMI USING w2v-BERT-L MODEL

MSM arch.	Type	IHM-eval	SDM-eval
w2v-BERT-L	Baseline	13.38	31.63
	Baseline+S	13.14	28.16
	ATM	12.52	27.34
	ATM+S	13.05	27.19

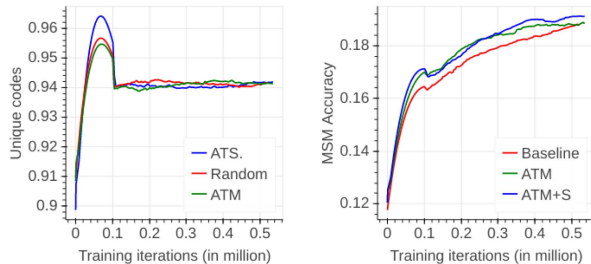


Fig. 6. Validating baseline, ATM and ATM+S using contrastive loss, number of unique codes used from the codebook and the MSM accuracy on dev-other during pre-training. A small bump is observed at around 0.1 million training iteration due to the change in learning rate and is ignored during analysis. .

evaluation sets using any of the MSM architectures. Slight gains are observed on test or dev or dev-other using w2v2-cfr-XL. Once the baseline in w2v-BERT-XL gets better, ATM did not achieve gains on test-other. This scenario can be explained due to effectiveness of MSM pre-training under matched condition and can perform well without any necessary data selection approach.

F. ATM With Loss Scaling (ATM+S) Analysis

ATM training can incorporate utterance-level weighting by scaling the MSM loss obtained using w2v-BERT-L models with the utterance-level confidence score according to Equation 6.

We evaluate the value of utterance-level loss scaling by re-weighting utterances in the context of both baseline MSM (i.e., without ATM frame selection) and ATM (ATM+S). These results are in Table IV. Re-weighting utterances by scaling the MSM loss with the confidence score on baseline model is denoted as “Baseline+S” and on ATM is labeled as “ATM+S”. MSM loss scaling is effective even without ATM; baseline+S improves over baseline on both IHM-eval and SDM-eval. Moreover, ATM+S improves over ATM on SDM-eval by attaining 27.19% WER while showing degradation on IHM-eval. This shows that ATM+S is effective on very hard evaluation task such as SDM-eval compared to IHM-eval. On the IHM-eval test set, the impact of MSM loss scaling is observed over the Baseline MSM without ATM. We hypothesize that ATM+S may not be able to provide improvement on IHM-eval as ATM already incorporates optimally incorporates scorer information on this task.

G. ATM+S Analysis on Validation Data During PT

The effect of ATM and ATM+S is analysed by plotting the validation scores on dev-other during pre-training. The first plot in Fig. 6 shows that the contrastive loss improves over the baseline with the aid of ATM and is further enhanced with

TABLE V
%WER ON LIBRISPEECH EVALUATION SETS USING ATM WITH UTTERANCE SCALING AND FRAME SCALING. THE FRAME SCALING IS ANALYSED WITH CHOOSING THE BEST PERCENTAGE OF UTTERANCES THAT PARTICIPATE IN SCALING.

Model	Type	PT-LL, FT-LS100			
		dev	dev-other	test	test-other
w2v-BERT-L	Baseline	3.78	8.86	3.85	9.32
	None	3.71	8.97	3.89	8.92
ATM	Utterance	3.64	8.79	3.95	8.95
	Frame-10%	3.73	8.97	3.84	9.06
	Frame-50%	4.01	9.35	4.14	9.54
	Frame-100%	4.23	9.77	4.89	9.97

TABLE VI
#WORD RECOVERED ON LIBRISPEECH EVALUATION SETS UWITH 0.1% ABSOLUTE IMPROVEMENT

Dataset	%WER Imp.	# Total words	# Words recovered (\approx)
dev-clean	0.1	54402	54
test-clean	0.1	52576	52
AMI-IHM	0.1	89635	89

ATM+S. The second plot shows the number of unique codes used from the quantizer codebook. Analysing this plot helps us to understand if the validation loss or accuracy is improved by just using less % of unique codes which will affect the performance at FT. Among the 1024 codes, 94%-95% are used by both ATM and ATM+S. This is similar to the % unique codes used by the baseline model and confirms that improvement of ATM and ATM+S is not by choosing smaller set of unique codes. The third plot shows that the MSM accuracy of ATM and ATM+S improves over baseline model. ATM+S shows that re-weighting each utterance is complementary to the ATM.

H. Comparison Between Frame-Level and Utterance-Level Loss Scaling

ATM+S performs MSM loss scaling using the utterance-level confidences which performs focus on each utterance at a coarse level. We also experimented with scaling with frame-level confidence scores. Our experiments showed that scaling all utterances with frame-level confidence hurts the model performance. To solve this issue, we randomly selected utterances which participate in frame scaling. Scaling 10% utterances resulted in better performance and the results are shown in Table V.

I. Statistical Significance Analysis on %WER Performance for Multiple Evaluation Sets

The Table VII results are on Librispeech and obtaining 0.1% improvement in Librispeech testsets is statistically significant. For instance, the dev-clean test set contains 54402 words and 0.1% gains denotes a recovery of 54 words. Also, the recent works on self-supervised training such as HuBERT shows improvement between wav2vec2-XL and HuBERT-XL only on dev-clean with 0.1% gain in Table III.

TABLE VII

%WER OBTAINED BY FT WITH LS-960 USING w2v-BERT-XL MODEL USING BASELINE, ATM AND ATM+S. THE RESULTS SHOW THE IMPACT OF OUR PROPOSED APPROACH ON MATCHED CONDITION SINCE LIBRISPEECH EVALUATION SETS ARE TREATED AS CLOSER TO LIBRI-LIGHT PT DOMAIN

MSM arch.	Type	dev	dev-other	test	test-other
w2v2-cfr-XL	Baseline [9]	1.7	3.5	1.7	3.5
w2v-BERT-XL	Baseline [9]	1.5	2.9	1.5	2.9
	ATM	1.4	2.8	1.5	2.9
	ATM+S	1.5	2.8	1.5	2.9

TABLE VIII

%WER OBTAINED BY FT WITH AMI USING w2v-BERT-XL MODEL USING BASELINE, ATM AND ATM+S. EVALUATION IS DONE ON AMI TEST SETS TO HIGHLIGHT THE EFFECT ON MISMATCHED CONDITION.

MSM arch.	Type	IHM-eval	SDM-eval
w2v2-cfr-XL	Baseline	10.4	25.7
	ATM	10.0	24.5
	ATM+S	9.8	23.9
w2v-BERT-XL	Baseline	10.1	25.1
	ATM	9.5	23.7
	ATM+S	9.5	23.5

VII. RESULTS

In this Section, XL models are used to compare the importance of ATM on LS-960, AMI and SpeechStew. These three datasets show the effect of ATM on diverse conditions with a much larger model. Results are compared with appropriate prior work.

Table VII shows that ATM and ATM+S improves over dev-other while on dev set there was improvement only using ATM and not ATM+S. Although the ATM and ATM+S does not show improvement on test and test-other, matches the very strong baseline. Considering the similarity between LS-960 and PT data, ATM manages to provide gains without hurting the performance across all Librispeech evaluation sets. This validates our argument that MSM models are better without any data selection when trained under matched data condition but can benefit under mismatched conditions.

Table VIII presents the results of ATM on AMI by comparing it with w2v2-conformer-XL baseline and w2v-BERT-XL baselines. We include w2v2-conformer-XL to further test the consistency of ATM on XL models when evaluated on harder tasks. ATM+S and ATM observes consistent gains over baseline on both IHM-eval and SDM-eval when trained with XL models. However, ATM+S did not demonstrate improvement on IHM-eval using w2v-BERT-XL.

Table IX analyses the effect of ATM and ATM+S on multiple evaluation sets such as Commonvoice, Tedlium, AMI and CHiME-6. These four sets are chosen based on the mismatch range from minimum to maximum and for instance, Commonvoice has the minimum mismatch with Libri-light data, while CHiME-6 has the maximum mismatch. The state-of-the-art results published in [11] are obtained by choosing the best Conformer model supervisedly trained with multiple datasets such as AMI, CommonVoice, Broadcast News, Librispeech, Switchboard/Fischer, TED-LIUM and Wall Street Journal. Note that the training data did not include the CHiME-6 data. The authors in [11] show that simply training an ASR with lots of

TABLE IX

COMPARISON WITH STATE-OF-THE-ART RESULTS ON SPEECHSTEW. THE FT IS DONE ON SPEECHSTEW AND THE RESULTS ARE EVALUATED USING KALDI SCORING TO MATCH PUBLISHED RESULTS. NOTE THAT THE MODEL HAS NEVER SEEN ANY CHiME-6 DATA, AND WE USE IT AS AN EXAMPLE FOR ZERO-SHOT LEARNING MODE.

Model Type	Common Voice	TED	AMI		CHiME-6
			IHM	SDM	
Speechstew [11]	12.1	5.3	9.0	21.7	57.2
w2v2-cfr-XL [11]	11.5	5.6	9.6	23.8	56.4
w2v-BERT-XL	11.2	5.3	9.2	23.6	55.5
+ ATM	10.8	5.3	9.0	21.0	54.3
+ ATM+S	10.7	5.2	8.9	20.7	53.9

data leads to best results compared to the wav2vec2 finetuned model. Their best results are denoted in Table IX and will be used to compare with our best ATM results.

Our baseline w2v-BERT-XL attained better results over the published w2v2-conformer-XL and Speechstew results. In Commonvoice and CHiME-6, the baseline attained 7.4% and 2.9% relative improvement over Speechstew respectively. However, by including our ATM and ATM+S with w2v-BERT-XL, there was consistent improvement across all range of mismatched domains. For instance, ATM+S attains 5.76% relative improvement on CHiME-6 over the Speechstew. This result clearly justifies that selection of reasonable input samples during pre-training reduces the necessity of having finetuning data from the same domain to improve performance. To further substantiate this, the results on AMI show a 4.6% relative improvement on AMI-SDM over Speechstew which is of different domain compared to pre-training domain. In case of minimal mismatch domain such as Commonvoice, the ATM attained 11.6% relative improvement over Speechstew. These observations show that ATM and ATM+S demonstrate their effectiveness to generalize to unseen and challenging speech recognition conditions.

VIII. CONCLUSION

In this work, we introduce ask2mask (ATM) to perform data selection over unsupervised samples for MSM based pre-training to focus on relevant information and learn meaningful representations. ATM achieves 21.0% WER on mismatched AMI SDM set with guided masking and a 20.7% WER is obtained by including loss scaling (ATM+S). We empirically show that ATM is more robust to changes in masking percentage compared to random masking, as typically used in MSM. Our results substantiate the importance of learning from high confident frames by attaining improvements across multiple evaluation sets. An important aspect of ATM approach is its flexibility to incorporate into any MSM pretraining techniques and ATM+S can also be easily adopted into self-supervised pre-training methods. In our future work, we wish to apply ATM over pretraining data containing data from multiple domains [10], [31] to achieve further improvements. We also consider two future enhancements to ATM: (1) Joint training of the scorer model with MSM model by simultaneous training on supervised and unsupervised data. (2) Perform active learning by sharing

TABLE X
COMPARISON BETWEEN SCORERS TRAINED WITH DIFFERENT AMOUNT OF SUPERVISED DATA (1 H, 10, 100 H OF LIBRISPEECH). THE SCORER IS EVALUATED ON 1 H OF LIBRI-LIGHT DATA.

#Hours.	% WER
1	45.4
10	27.6
100	25.8

TABLE XI
%WER BETWEEN THREE ATM MODELS PRETRAINED WITH LL-60 K DATA AND CORRESPONDING THREE SCORERS (1 H, 10 AND 100 H). THE ABOVE THREE MODELS ARE FT WITH 100 HOURS OF AMI AND EVALUATED WITH IHM-EVAL AND SDM-EVAL.

ATM Scorer	IHM-eval	SDM-eval
Scorer-1h	9.8	24.1
Scorer-10h	9.6	23.7
Scorer-100h	9.5	23.7

TABLE XII
%WER COMPARISON BETWEEN ATM MODEL TRAINED WITH SCORER WITH AND WITHOUT DECODER. THE FT IS DONE WITH AMI USING ATM

Scorer.	Modules used	IHM-eval	SDM-eval
Supervised	Enc+Dec	9.5	23.7
	Enc	9.1	23.6
Unsupervised	Enc	9.4	23.8

the parameters of MSM with the scorer once the MSM is well trained.

APPENDIX

A. Analysis of Amount of Supervised Data for Scorer

We build three different scorers using 1 h, 10 hours and 100 hours of Librispeech data. The scorers are evaluated using 1 h of Libri-light data and the results are in Table X. We chose 1 h of Libri-light data since the pretraining data is from the same domain (Libri-light).

Table XI shows that the amount of supervised data used to train the scorer has negligible impact on ATM performance after 10 hours of supervision. Scorer trained with 1 h of supervised data performed relatively poor compared to 10 hours and 100 hours scorer. The pattern is similar to the performance improvements evaluated on 1 h of Libri-light data.

B. Unsupervised Scorer Analysis

In this work, we chose to use the target domain data used during finetuning to train the scorer model without increasing the amount of supervised data. Our later experiments with scorer is to use the unsupervisedly available target domain data and compare its performance with supervised scorer. We performed two sets of experiments to study the impact of unsupervised scorer. First, we removed the CTC decoder from the supervised scorer and softmax over the encoder representations to understand the importance of decoder in scorer model. Table XII shows that removing the CTC layer from the scorer led to better selection of acoustic frames without depending on the text sequence information from decoder.

TABLE XIII
%WER COMPARISON BETWEEN ATM MODEL TRAINED WITH DIFFERENT RANDOM SEEDS DUEING PRETRAINING. THE RESULTING MODELS ARE FINETUNED WITH AMI

Seed	IHM-eval	SDM-eval
1234 (default)	12.52	27.34
4321	12.50	27.33
603	12.51	27.35
37	12.54	27.35

With this motivation, we trained an encoder only model (100 M parameters) using w2v-BERT objective with 100 hours of AMI training data. The encoder output has 256 dimensions and are fed to softmax layer to obtain confidence scorer. This acts as an unsupervised scorer in Table XII and shows comparable performance to supervised scorer. This finding highlights that the latent representations from the unsupervised scorer is helpful to perform better data selection.

C. ATM Pretraining With Random Seeds

The ATM pretraining is critically analysed by changing the seed set for weight initialization. The random seeds are selected by sampling from random number generator. The results in Table XIII shows that performance of ATM remains similar across different initializations on both IHM-eval and SDM-eval test sets.

ACKNOWLEDGMENT

We would like to thank Lukáš Burget, Jan Honza Černocký, Bolaji Yusuf and Zhehuai Chen for helping to review the paper and provide valuable suggestions.

REFERENCES

- [1] H. Scudder, "Probability of error of some adaptive pattern-recognition machines," *IEEE Trans. Inf. Theory*, vol. 11, no. 3, pp. 363–371, Jul. 1965.
- [2] J. Kahn, A. Lee, and A. Hannun, "Self-training for end-to-end speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 1330–1334.
- [3] D. S. Park et al., "Improved noisy student training for automatic speech recognition," in *Proc. Interspeech*, 2020, pp. 2817–2821, doi: [10.21437/Interspeech.2020-1470](https://doi.org/10.21437/Interspeech.2020-1470).
- [4] Q. Xu et al., "Self-training and pre-training are complementary for speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 1330–1334.
- [5] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length, and Helmholtz free energy," in *Proc. 6th Int. Conf. Neural Inf. Process. Syst.*, 1994, pp. 3–10.
- [6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, vol. 1, Jun. 2019, pp. 4171–4186, doi: [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423).
- [7] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "Wav2Vec 2.0: A framework for self-supervised learning of speech representations," *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 12449–12460. [Online]. Available: <https://proceedings.neurips.cc/paper/2020/file/92d1e1eb1cd6f9fba3227-870bb6d7f07-Paper.pdf>
- [8] W.-N. Hsu, Y.-H. H. Tsai, B. Bolte, R. Salakhutdinov, and A. Mohamed, "Hubert: How much can a bad teacher benefit ASR pre-training?," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 1330–1334.
- [9] Y.-A. Chung et al., "W2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," *IEEE Autom. Speech Recognit. Understanding Workshop (ASRU)*, 2021, arXiv:2108.06209.

- [10] W.-N. Hsu et al., "Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training," in *Proc. Interspeech*, 2021, pp. 721–725, doi: [10.21437/Interspeech.2021-236](https://doi.org/10.21437/Interspeech.2021-236).
- [11] W. Chan, D. S. Park, C. Lee, Y. Zhang, Q. V. Le, and M. Norouzi, "Speechstew: Simply mix all available speech recognition data to train one large neural network," 2021, *arXiv:2104.02133*.
- [12] Y. Sun et al., "ERNIE: Enhanced language representation with informative entities," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2019, pp. 1441–1451, doi: [10.18653/v1/P19-1139](https://doi.org/10.18653/v1/P19-1139).
- [13] C. Wang et al., "Semantic mask for transformer based end-to-end speech recognition," in *Proc. Interspeech*, 2020, pp. 971–975, doi: [10.21437/Interspeech.2020-1778](https://doi.org/10.21437/Interspeech.2020-1778).
- [14] X. Yue and H. Li, "Phonetically motivated self-supervised speech representation learning," in *Proc. Interspeech*, 2021, pp. 1330–1334.
- [15] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1330–1334.
- [16] K. Veselý, L. Burget, and J. Cernocký, "Semi-supervised DNN training with word selection for ASR," in *Proc. Interspeech*, 2017, pp. 1330–1334.
- [17] J. Ferreira, M. Mendonca, and P. S. Diniz, "Data selection in neural networks," *IEEE Open J. Signal Process.*, vol. 2, pp. 1330–1334, 2021.
- [18] F. Wessel, R. Schluter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 3, pp. 288–298, Mar. 2001.
- [19] Z. Ren, R. Yeh, and A. Schwing, "Not all unlabeled data are equal: Learning to weight data in semi-supervised learning," *Proc. Conf. Neural Inf. Process. Syst.*, no. 33, pp. 21786–21797, 2020.
- [20] C. Coleman et al., "Selection via proxy: Efficient data selection for deep learning," 2020. [Online]. Available: <https://openreview.net/forum?id=HJg2b0VYDr>
- [21] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: Improving pre-training by representing and predicting spans," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 64–77, 2020.
- [22] S. Khurana, N. Moritz, T. Hori, and J. Le Roux, "Unsupervised domain adaptation for speech recognition via uncertainty driven self-training," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 1330–1334.
- [23] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1330–1334.
- [24] Y. Levine et al., "PMI-masking: Principled masking of correlated spans," 2020, *arXiv:2010.01825*.
- [25] Y. Zhang et al., "Pushing the limits of semi-supervised learning for automatic speech recognition," 2020, *arXiv:2010.10504*.
- [26] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2015, pp. 1330–1334.
- [27] S. W. et al., "CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings," in *Proc. 6th Int. Workshop Speech Process. Everyday Environ. (CHiME 2020)*, 2020, pp. 1–7. [Online]. Available: <https://doi.org/10.21437/CHiME.2020-1>
- [28] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 1330–1334.
- [29] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1330–1334.
- [30] Q. Zhang et al., "Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 1330–1334.
- [31] T. Likhomanenko et al., "Rethinking evaluation in ASR: Are our models robust enough?," in *Proc. Interspeech*, 2021, pp. 311–315. [Online]. Available: <https://doi.org/10.21437/Interspeech.2021-1758>