Proceedings Paper:
Park, C. orcid.org/0000-0001-6671-1671, Lu, C., Chen, M. et al. (1 more author) (2025) Fast word error rate estimation using self-supervised representations for speech and text. In: ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 06-11 Apr 2025, Hyderabad, India. Institute of Electrical and Electronics Engineers (IEEE) , pp. 1-5. ISBN 9798350368758

https://doi.org/10.1109/icassp49660.2025.10890056

FAST WORD ERROR RATE ESTIMATION USING SELF-SUPERVISED REPRESENTATIONS FOR SPEECH AND TEXT

Chanho Park, Chengsong Lu, Mingjie Chen, Thomas Hain
Computer Science Department, University of Sheffield
Speech and Hearing Research Group
Sheffield, UK
cpark12, clu22, mingjie.chen, t.hain@sheffield.ac.uk

***Abstract*—Word error rate (WER) estimation aims to evaluate the quality of an automatic speech recognition (ASR) system without requiring ground truth labels. This task has gained increasing attention as advanced ASR systems are trained on large amounts of data. In this context, the computational efficiency of a WER estimator becomes essential in practice. However, previous works have not prioritised this aspect. In this paper, a Fast estimator for WER (Fe-WER) using a self-supervised learning representation (SSLR) is introduced. The estimator employs average pooling over SSLR. Our results demonstrate that Fe-WER outperformed a baseline relatively by 14.10% in root mean square error and 1.22% in Pearson correlation coefficient on Ted-Lium3. Moreover, a comparative analysis of the distributions of target WER and WER estimates was conducted, including an examination of the average values per speaker. Lastly, the inference speed was approximately 3.4 times faster in the real-time factor.**

***Index Terms*—Word error rate, WER estimation, self-supervised representation, multi-layer perceptrons, inference speed**

I. Introduction

Word error rate (WER) is a commonly used metric for evaluating automatic speech recognition (ASR) systems. It is the ratio of the number of substitution, insertion, and deletion errors in a hypothesis to the number of words in a reference. In some scenarios, it can be very useful to use a model to estimate the WER of an ASR system's output, especially when the ground-truth transcript is not available. For example, a WER estimation model can be used to rank hypotheses [13] and to select unlabelled data for ASR self-training [3], [15], [25]. Another use may be to filter out training data with high-WER transcripts, especially when they are collected from the internet. To achieve good ASR performance, data samples with high-WER transcripts usually need to be excluded from ASR training, particularly for recent ASR models, e.g., Whisper [19], that are trained with large amounts of data collected from the internet. When dealing with large amounts of data, the computational efficiency of a WER estimator becomes important. One obvious solution to estimate the WER of an ASR system's output is to produce confidence scores from the ASR system itself [14], [16]. This method does not require building another model for WER. However, this has the risk of bias and—as will be shown—does not perform well compared to WER estimation methods. Additionally, it is not aligned with WER due to the lack of prediction of deletion errors.

Recently, researchers have proposed methods to directly estimate the WER of an ASR system's output without the need for ASR decoding. For example, e-WER3 [4] used bidirectional long short-term memory (BiLSTM) networks to extract features for speech, while the features for text were averaged over tokens. Then, WER was directly estimated using multi-layer perceptrons (MLP) with these features. Although it has made impressive progress in estimating the WER of ASR systems, there are still several questions that have not been fully studied. Firstly, the e-WER3 model, though avoiding ASR decoding, relies on BiLSTMs, which are computationally intensive for long sequences like spoken utterances. This limits their use in training with long speech. Secondly, the performance of the estimator depends on the input features for speech and text. Thus, different combinations of self-supervised learning representations (SSLRs) for speech and text need to be explored for optimal performance on the WER estimation task. Lastly, performance needs to be analysed across data attributes, such as utterance lengths and speakers in addition to the evaluation metrics.

In this paper, a framework to build a Fast estimation model for WER (Fe-WER) consisting of speech and text encoders, feature aggregators and a WER estimator, is proposed. The SSLRs aggregated by average pooling are used to directly estimate WER with MLP. This framework will be explored from accuracy and efficiency perspectives. The contributions of this paper are as follows:

1) This paper proposes a WER estimation model using average pooling, Fe-WER, which outperforms the baseline model in computational efficiency without performance degradation.
2) Experimental evidence shows that the combination of HuBERT [11] and XLM-R [6] achieves the best performance in WER estimation.
3) A comparative analysis of the distributions of target WER and WER estimates is presented including an examination of the average values per speaker.

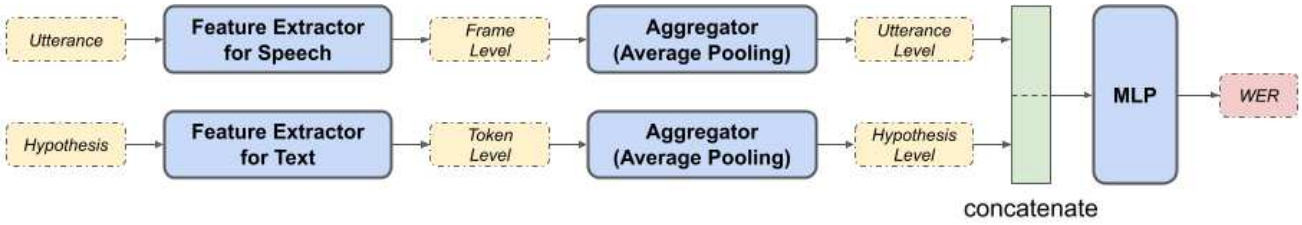


Fig. 1: Illustration of the proposed method for WER estimation

II. Related works

A. WER Estimation

e-WER3 is a WER estimator for multiple languages. In [4], hypotheses were generated by a conformer-based ASR system [8] trained on LibriSpeech [18]. The features of utterances and hypotheses were extracted using XLSR-53 [5] and XLM-R [6]. The hidden states of a BiLSTM in both directions over frame-level representations were concatenated to form an utterance-level representation, while an transcript-level representation was averaged over token-level representations. For data selection, hypotheses whose WER was equal to 0 were selected up to the sum of the numbers in the second and third most frequent groups. The WER was predicted using fully connected layers on top of the concatenated representation. The result was 0.14 in root mean square error (RMSE) and 0.66 in Pearson correlation coefficient (PCC) on the English corpus, Ted-Lium3 [10], which was improved relatively by 9% in PCC from e-WER2.

B. Sequence-level Representation

In [21], a sentence-level representation was suggested for NLP tasks, such as semantic textual similarity (STS) between sentences. The representation, called SBERT, was learned using a Siamese or a triplet model—often referred to as a two-tower architecture [12], [26]—with classification, regression and triplet objective functions. One of the SSLRs, BERT [7], was adopted and converted into a fixed-length representation for a sentence through different pooling strategies. The results showed that the average pooling strategy outperformed the others, such as using a special token for classification of BERT. In addition to SBERT, the average pooling strategy for utterance-level representation has gained popularity in many other tasks, such as speaker identification, intent classification and emotion recognition [23], [24].

III. Fast Word Error Rate Estimation

A. Architecture

Fe-WER (see Fig. 1) is based on a two-tower architecture, which maps different representations into a shared space to capture the similarity between two inputs. The proposed model consists of two aggregators—one for speech and another for text—and fully connected layers that perform the WER estimation. The aggregators convert the features extracted by SSLRs into sequence-level representations. These two sequence-level representations are concatenated and input to an MLP consisting of fully connected layers with a rectified linear unit (ReLU) activation function. A sigmoid function is applied to the output. The WER estimate $\widehat{WER}$ is defined:

$$\widehat{\text{WER}} = \text{MLP}(\text{concat}(p(f(s)), p(g(t))))$$

where p is a function of average pooling, $f(\cdot)$ and $g(\cdot)$ are encoders for speech and text, respectively, and s and t are a spoken utterance and an automatic transcript, respectively.

B. Training Objective

The mean squared error (MSE) between WER and $\widehat{\text{WER}}$ is used as the objective function to train the MLP, where WER represents the error rate between references and hypotheses and $\widehat{\text{WER}}$ is the estimation by the model.

$$\text{MSE} = \frac{\sum_{i=1}^{N}(\text{WER}_i - \widehat{\text{WER}}_i)^2}{N}$$

where N is the number of instances in a dataset and i is the index of an instance.

C. Weighted Word Error Rate Estimate

The word error rate can be weighted by the number of words in a reference transcript, denoted as WER_{wrd}. For the weighted WER estimation on a dataset, it is weighted by duration instead of the number of words in the reference. The weighted WER estimate is defined as follows:

$$\widehat{\text{WER}}_{\text{dur}} = \frac{\sum_{i=1}^{N}(\widehat{\text{WER}}_i \times \text{Duration}_i)}{\sum_{i=1}^{N}(\text{Duration}_i)}$$

where i is the index of a pair consisting of an utterance and its corresponding hypothesis.

D. Evaluation Metrics

PCC and RMSE are used as evaluation metrics. A PCC value close to 1 indicates that two variables change in the same direction, while a value close -1 indicates that they change in opposite directions.

$$\frac{\sum_{i=1}^{N}(\text{WER}i - \mu_{\text{WER}})(\widehat{\text{WER}}_i - \mu_{\widehat{\text{WER}}})}{\sqrt{\sum_{i=1}^{N}(\text{WER}i - \mu_{\text{WER}})^2 \sum_{i=1}^{N}(\widehat{\text{WER}}_i - \mu_{\widehat{\text{WER}}})^2}}$$

where μ_{WER} is the mean of WER. Lastly, the ratio between the weighted WER_{wrd} and $\widehat{\text{WER}}_{\text{dur}}$ is also measured.

$$\text{WERR} = \frac{|\text{WER}_{\text{wrd}} - \widehat{\text{WER}}_{\text{dur}}|}{\text{WER}_{\text{wrd}}}.$$

TABLE I: Statistics of the sets of data selected. Hypotheses were generated by Whisper large-v2.

Dataset	#seg.	total dur. (h)	avg. dur.	avg. #wrd.	avg. WER	std. dev. of WER	WER_{wrd}
eval	842	1.41	6.05	16.72	0.1429	0.1997	0.1088
dev	1034	1.70	5.93	17.72	0.1532	0.2247	0.1225
train	123255	200.55	5.86	17.04	0.2434	0.3209	0.2029

IV. Experiment Setup

A. Data

TED-LIUM3 (TL3) [10] was used as the ASR corpus for WER estimation. For transcribing the corpus, Whisper large-v2 [1] was employed for reproducibility, as it demonstrated comparable performance on TL3 and is publicly available. The transcribed data were highly imbalanced due to the high volume of WER 0. To address this issue, hypotheses with a WER of 0 were filtered out based on the WER distribution, as described in Section II-A. Additionally, utterances with lengths up to 10 seconds were selected, and WER was clamped between 0% and 100%. Whisper's text normaliser was modified to prevent the replacement of numeric expressions with Arabic numerals. The statistics of the selected data are summarised in Table I.

B. Self-supervised Learning Representations

SSLRs for utterances and hypotheses were selected based on their performance on benchmarks including Speech processing Universal PERformance Benchmark (SUPERB) [24], General Language Understanding Evaluation (GLUE) [23] and SuperGLUE [22]. These benchmarks assess models on various tasks, such as phoneme recognition and paraphrase detection. Additionally, two models used for WER estimation with BiLSTM [4] were included. Summary information on these models, including model size and the number of parameters, is provided in Table II.

TABLE II: Summary information of SSLRs.

Type	Model	Abbr.	Size	#Parameters
Utterance	data2vec [1]	DA	Large	313M
	HuBERT [11]	HU	Large	316M
	WavLM [2]	WA	Large	317M
	XLSR-53 [5]	XS	Large	315M
Transcript	DeBERTa-V3 [9]	DE	Large	283M
	GPT-2 [20]	GP	Medium	355M
	RoBERTa [17]	RO	Large	355M
	XLM-R [6]	XM	Large	560M

C. Baseline WER Estimators

The proposed model was compared with two baselines: a method using a confidence score (WER-CS) and another with BiLSTM. First, for sequence-level confidence scoring, the log probability of Whisper large-v2 over the output tokens was averaged and subtracted from 1. For decoding, two strategies were employed: greedy decoding only and full decoding. The full decoding strategy included a beam size of 5, greedy decoding with the 5 best hypotheses and sampling temperature settings ranging from 0 to 1 in increments of 0.2. Second, another method using BiLSTM was implemented for the second baseline. A single-layer BiLSTM was used to aggregate SSLR representations, with the input and hidden feature sizes matching the size of the inputs. For further details, readers can refer to e-WER3 [4].

[1]https://github.com/openai/whisper

D. Fe-WER

Average pooling over the frame or token dimension was used as the aggregator. Hyperparameters were selected via grid search. The Fe-WER includes an MLP with 2 hidden layers and 1 output layer, activated by ReLU and Sigmoid functions, respectively. Each layer's output is normalised, and dropout (0.1) is applied to the hidden layers. The fully connected layers consist of 3 layers with 600, 32, and 1 nodes on top of 2048-dimensional input features. The model was trained with an Adam optimiser (learning rate: 1e-3), a cosine annealing scheduler and early stopping at 40 epochs.

V. Results

First, BiLSTM and average pooling are compared across different combinations of SSLRs. Next, the WER estimation models with the best SSLR combination are compared with a baseline using confidence scoring. This is followed by an analysis of WER estimation at the utterance level and a comparison of inference speed.

A. Aggregators

Aggregators using BiLSTM and average pooling were compared with combinations of SSLRs in Section IV-B. First, RMSE and PCC tend to improve with average pooling in 13 out of 16 combinations. Second, the best combinations are DA and XM for BiLSTM and HU and XM for average pooling. The latter outperformed the former by 0.0099 in RMSE and 0.0228 in PCC on TL3 dev. Results are summarised in Table III.

B. Comparison with Baselines

The proposed model, which uses an average pooling aggregator with HU and XM, is compared to two baselines: WER-CS and a model using BiLSTM with DA and XM. First, the two decoding strategies described in Section IV-C were applied to WER-CS with Whisper large-v2. However, it performed worse than the other models in both metrics with all strategies. The proposed model outperformed the baselines in RMSE and PCC by at least 14.10% and 1.22%, respectively. The comparison results are shown in Table IV.

TABLE III: Results of BiLSTM and Average pooling aggregators with different SSLRs combinations on TL3 dev. All models were trained with three seeds.

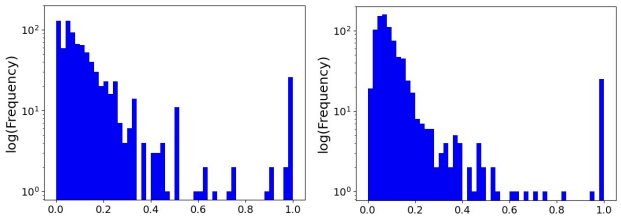
SSLR		BiLSTM		Average Pooling	
Utt.	Hyp.	RMSE↓	PCC↑	RMSE↓	PCC↑
DA	DE	.1185±.001	.8490±.004	.1213±.000	.8425±.001
DA	GP	.1254±.005	.8405±.008	.1185±.001	.8512±.002
DA	RO	.1193±.002	.8491±.008	.1190±.002	.8486±.004
DA	XM	**.1111±.008**	**.8700±.018**	.1137±.001	.8637±.002
HU	DE	.1216±.002	.8398±.004	.1105±.002	.8702±.005
HU	GP	.1233±.002	.8387±.005	.1093±.001	.8741±.001
HU	RO	.1227±.004	.8363±.011	.1123±.003	.8676±.006
HU	XM	.1212±.011	.8418±.032	**.1012±.003**	**.8928±.007**
WA	DE	.1289±.005	.8200±.014	.1164±.002	.8551±.003
WA	GP	.1270±.003	.8245±.009	.1111±.002	.8709±.006
WA	RO	.1210±.004	.8420±.013	.1167±.002	.8561±.004
WA	XM	.1172±.005	.8520±.015	.1099±.002	.8734±.005
XS	DE	.1289±.003	.8191±.011	.1216±.002	.8412±.006
XS	GP	.1200±.003	.8467±.008	.1155±.001	.8585±.002
XS	RO	.1285±.003	.8226±.006	.1161±.003	.8567±.007
XS	XM	.1199±.005	.8474±.009	.1101±.001	.8717±.003

TABLE IV: RMSE and PCC of baseline systems on TL3 eval. WER_{wrd} is a target WER weighted by words. $\widehat{\text{WER}}_{\text{dur}}$ is the WER estimate weighted by duration. † is the proposed method.

	RMSE↓	PCC↑	WER_{wrd}	$\widehat{\text{WER}}_{\text{dur}}$	WERR↓
WER-CS					
+ full	0.2611	0.5654	8.40%	31.85%	2.7916
+ greedy	0.2546	0.6944	10.88%	33.34%	2.0643
BiLSTM					
+ DA,XM	0.1071	0.8793	10.88%	10.96%	0.0073
†Avg. Pool.					
+ HU,XM	0.0920	0.8900	10.88%	10.39%	0.0450

C. Distributions of Target WER and WER Estimates

The histograms of target WERs and WER estimates on TL3 eval are visualised in Fig. 2. The distribution of Fe-WER estimates is similar to that of the target WERs. However, the frequency of target WERs peaks in the [0.00, 0.02) range in Fig. 2(a), while the estimates peak in the [0.04, 0.08) range in Fig. 2(b). This discrepancy may be due to the Sigmoid function outputting small values rather than 0. Additionally, WER estimates around 0.2 are less frequent than target WERs. In this range, three or more insertions in a row are frequently observed in the hypotheses. Recognising these words as one insertion error could have led to the low estimates.



(a) target WER (b) WER estimate

Fig. 2: Histograms on TL3 eval

D. Average Target WER and WER Estimate per Speaker

The distributions of average target WER and WER estimate per speaker are similar (See Fig. 3). The high average target WER of Speaker 5 is due to majority of shorter utterances, which have low resolution of WER. For example, the WER of a spoken utterance for a word is 0 or at least 100%. For Speaker 16, the average WER estimate is higher than the average WER target. The phenomenon of high WER estimate was discussed in Section V-C.

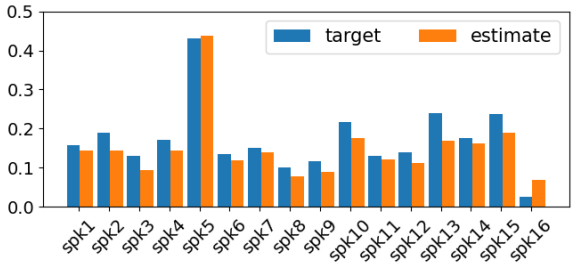


Fig. 3: Average WER per each speaker

E. Inference Speed

The inference time of the WER estimators was measured on a single NVIDIA RTX A6000 GPU with a batch size of 1, including the encoding time. The model using BiLSTM had an inference time of 18.64 seconds, while the proposed method's inference time was significantly shorter at 5.42 seconds, reducing the inference time by approximately 70.92%. The details are summarised in Table V.

TABLE V: RTF of BiLSTM and Avg. Pool. with HU and XM on TL3 eval. Total duration is about 5223 seconds. RTF: total time ÷ total duration. † is the proposed method.

	BiLSTM	†Avg. Pool.
Feature extraction		
+ utterance		2.72
+ transcript		0.93
Aggregation	5.28	ϵ
Feedforward	9.71	1.77
Total	18.64	5.42
RTF	0.003569	0.001038

VI. Conclusion

In this paper, a Fast WER estimator is proposed. The proposed model consists of SSLR encoders for speech and text, aggregators using average pooling and an MLP estimator. The WER estimator outperforms the BiLSTM baseline by relative 14.10% and 1.22% in RMSE and PCC, respectively. Moreover, the experimental results show that the inference speed has been significantly improved, being 3.4 times faster than the BiLSTM baseline, without performance degradation.

Acknowledgement

This work was conducted at the Voicebase/Liveperson Centre of Speech and Language Technology at the University of Sheffield which is supported by Liveperson, Inc..

REFERENCES

[1] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language, 2022.

[2] Sanyuan Chen et al. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, July 2022.

[3] Yang Chen, Weiran Wang, and Chao Wang. Semi-supervised ASR by end-to-end self-training. In *Proc. Interspeech 2020*, pages 2787–2791, 2020.

[4] Shammur Absar Chowdhury and Ahmed Ali. Multilingual word error rate estimation: e-WER3. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023.

[5] Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. Unsupervised cross-lingual representation learning for speech recognition. In *Proc. Interspeech 2021*, pages 2426–2430, 2021.

[6] Alexis Conneau et al. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol., Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics.

[8] Anmol Gulati et al. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Proc. Interspeech 2020*, pages 5036–5040, 2020.

[9] Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving deberta using ELECTRA-style pre-training with gradient-disentangled embedding sharing, 2021.

[10] François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Estève. TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In Alexey Karpov, Oliver Jokisch, and Rodmonga Potapova, editors, *Speech and Computer*, pages 198–208, Cham, 2018. Springer International Publishing.

[11] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

[12] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, CIKM '13, page 2333–2338, New York, NY, USA, 2013. Association for Computing Machinery.

[13] Shahab Jalalvand, Matteo Negri, Daniele Falavigna, and Marco Turchi. Driving ROVER with segment-based ASR quality estimation. In Chengqing Zong and Michael Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1095–1105, Beijing, China, July 2015. Association for Computational Linguistics.

[14] Woojay Jeon, Maxwell Jordan, and Mahesh Krishnamoorthy. On modeling ASR word confidence. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6324–6328, 2020.

[15] Jacob Kahn, Ann Lee, and Awni Hannun. Self-training for end-to-end speech recognition. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7084–7088, 2020.

[16] Ankur Kumar, Sachin Singh, Dhananjaya Gowda, Abhinav Garg, Shatrughan Singh, and Chanwoo Kim. Utterance confidence measure for end-to-end speech recognition with applications to distributed speech recognition scenarios. In *Proc. Interspeech 2020*, pages 4357–4361, 2020.

[17] Yinhan Liu et al. RoBERTa: A robustly optimized BERT pretraining approach. Meta AI., https://ai.facebook.com/blog/ roberta-an-optimized-method-for-pretraining-self-supervised-nlp-systems (Accessed: Jun 22, 2022).

[18] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015.

[19] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[20] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[21] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.

[22] Alex Wang et al. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[23] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[24] Shu wen Yang et al. SUPERB: Speech Processing Universal PERformance Benchmark. In *Proc. Interspeech 2021*, pages 1194–1198, 2021.

[25] Qiantong Xu, Alexei Baevski, et al. Self-training and pre-training are complementary for speech recognition. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3030–3034, 2021.

[26] Ji Yang et al. Mixed negative sampling for learning two-tower neural networks in recommendations. In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 441–447, New York, NY, USA, 2020. Association for Computing Machinery.