Interspeech 2024
1-5 September 2024, Kos, Greece

Optimizing Automatic Speech Assessment: W-RankSim Regularization and Hybrid Feature Fusion Strategies

Chung-Wen Wu[1], *Berlin Chen*[1]

[1]Department of Computer Science and Information Engineering, National Taiwan Normal University

40947040s@ntnu.edu.tw, berlin@ntnu.edu.tw

Abstract

Automatic Speech Assessment (ASA) has seen notable advancements with the utilization of self-supervised features (SSL) in recent research. However, a key challenge in ASA lies in the imbalanced distribution of data, particularly evident in English test datasets. To address this challenge, we approach ASA as an ordinal classification task, introducing Weighted Vectors Ranking Similarity (W-RankSim) as a novel regularization technique. W-RankSim encourages closer proximity of weighted vectors in the output layer for similar classes, implying that feature vectors with similar labels would be gradually nudged closer to each other as they converge towards corresponding weighted vectors. Extensive experimental evaluations confirm the effectiveness of our approach in improving ordinal classification performance for ASA. Furthermore, we propose a hybrid model that combines SSL and handcrafted features, showcasing how the inclusion of handcrafted features enhances performance in an ASA system.

Index Terms: automatic speech assessment, ordinal classification, imbalanced data

1. Introduction

Many people learning English as a second language (L2) lack exposure to a native environment and may struggle to receive timely feedback from proficient English speakers. In such scenarios, ASA can readily provide a speech proficiency score, benefiting L2 learners in their practice [1, 2]. Additionally, it aids language tests by efficiently ensuring more consistent scoring.

Early studies in ASA [3, 4, 5, 6] primarily utilized handcrafted features associated with various aspects of language proficiency, including content (such as relevance), delivery (such as pitch, duration, silence), and language use (such as Part-of-Speech (POS) tags, syntactic dependencies (DEP), morphology), among others. While these features provide interpretability and are directly linked to human grading criteria, they inevitably face challenges in generalizability and are susceptible to errors during handcrafted feature extraction, especially when concerning features related to content and language use.

Recent research has highlighted the effectiveness of SSL across various speech processing tasks [7, 8, 9, 10, 11, 12], including but is not limited to automatic speech recognition (ASR), keyword spotting, mispronunciation detection and diagnosis (MDD), emotion recognition, and speaker diarisation. Pre-trained models such as wav2vec 2.0 [7], Whisper [13], and HuBERT [14] leverage large amounts of data and wherein the contextual information to extract more robust and generalized features. In ASA, these models have demonstrated effectiveness in representing high-level speech-related features and linguistic aspects [15] such as fluency, pronunciation, semantics and even syntax, thereby enhancing overall assessment performance.

A key challenge encountered in ASA is data imbalance, particularly prevalent in datasets sourced from English tests [16, 17]. In these tests, the score distribution tends towards a normal distribution, leading to a scarcity of data points for the lowest and highest scores. Recent research [4, 15, 16, 17, 18] often approaches the ASA task as either a regression or classification problem, utilizing mean square error loss or cross-entropy loss functions for training. However, both of these loss functions neglect the ordinal nature of the scores and are highly affected by data imbalance.

In this paper, we present a hybrid model that combines both pretrained model features and handcrafted features. We conduct a series of experiments and ablation studies to demonstrate how integrating handcrafted features enhances performance. Moreover, we treat the ASA as an imbalanced ordinal classification challenge and leverage the ordinal characteristics of the scores to enhance accuracy. To address this challenge, we put forward an effective optimization framework for ASA modeling, dubbed W-RankSim, which builds upon the concept of RankSim [19].

RankSim is implemented as a batch-wise approach, leveraging ordinal information in regression tasks. However, it often requires a large batch size to demonstrate improvements. In tasks such as ASA, where input signals are substantial, there may not be sufficient VRAM to accommodate such large batch sizes. Additionally, in ordinal classification tasks, RankSim encounters difficulties in accumulating gradients for each class in every batch, as it can only accumulate the gradient of labels in the batch. To address these limitations, we propose W-RankSim, which operates in the weighted vector space. W-RankSim overcomes the batch size constraint of RankSim, reducing training overhead and effectively accumulating gradients for each class in ordinal classification tasks. Furthermore, we identify a suitable loss function to complement W-RankSim.

In our experiments, we demonstrate that W-RankSim successfully overcomes the limitation of batch size in RankSim and achieves superior performance on ordinal classification tasks.

In summary, this paper presents three main contributions:

1. Suggesting to approach ASA as an imbalanced ordinal classification problem to enhance the performance by leveraging ordinal information.
2. Introducing an effective regularization to enhance predictive performance on imbalanced ordinal classification tasks.
3. Proposing a hybrid model to demonstrate the usefulness of handcrafted features in building an ASA system.

To the best of our knowledge, this paper is the first to define ASA as an imbalanced ordinal classification task and propose a pragmatic method to improve performance.

10.21437/Interspeech.2024-1078

2. Methodology

To address the data imbalance problem in ASA systems, which are typically scored using ordinal levels such as CEFR (Common European Framework of Reference for Languages) levels and scores in English tests, we treat the task as an ordinal classification problem. In this context, we proposed W-RankSim (Section 2.1) to leverage the ordinal information in the scores effectively. Additionally, we introduced a novel hybrid model (Section 2.3) to further enhance the performance of ASA systems.

There are some terminologies to clarify: The set of ordinal class labels is denoted by C. Our datasets consist of pairs (x_i, y_i), where x_i represents the input vector and $y_i \in C$ denotes the label. The final hidden feature space before predicting the class is denoted as Z, where $z_i \in Z$ is obtained by passing x_i through a neural network. The last weight matrix in the output layer is represented by $W \in \mathbb{R}^{|C| \times H^o}$, where H^o is the hidden dimension in output head.

Let w_j denote the weight vector in W, where its corresponding label is the j-th label. In other words, w_j represents the weight vector associated with the j-th class. Consequently, the predicted confidence score of z_i in the j-th class is computed as the dot product of w_j and z_i, denoted as $w_j \cdot z_i$.

2.1. W-RankSim

Consider an arbitrary vector $a \in \mathbb{R}^n$. Define **rk** as the ranking function, where $\mathbf{rk}(a)$ represents the permutation of $\{1, 2, \ldots, n\}$ containing the rank of each element in a. In other words, the ith element in $\mathbf{rk}(a)$ is expressed by

$$\mathbf{rk}(a)_i = 1 + |\{j : a_j > a_i\}| \tag{1}$$

Additionally, let σ^w denote the cosine similarity function. We apply cosine similarity to each pair of weighted vectors (w_i, w_j) to obtain the weighted vectors similarity matrix S^w. Each entry in $S^w \in \mathbb{R}^{|C| \times |C|}$ can be represented by

$$S^w_{i,j} = \sigma^w(w_i, w_j) \tag{2}$$

In the label space, each pair of class labels (c_i, c_j), where $c_i, c_j \in C$, are passed to σ^c,which is a simple negative absolute distance function, to construct $S^c \in \mathbb{R}^{|C| \times |C|}$:

$$S^c_{i,j} = \sigma^c(c_i, c_j) \tag{3}$$

Then, W-RankSim in ordinal class labels C is constructed as:

$$L_{\text{W-RankSim}} = \sum_{i=1}^{|C|} l\left(\mathbf{rk}(S^c_{[i,:]}), \mathbf{rk}(S^w_{[i,:]})\right) \tag{4}$$

where $[i, :]$ denotes the i-th column in the matrix, and l is a ranking similarity function that aims to make the rank similarity between the label space and weighted vector space similar. In this case, mean square error is used as l.

During training, the main loss function L_{main} is combined with W-RankSim. The complete formula is as follows:

$$L_{\text{total}} = L_{\text{main}} + \gamma L_{\text{W-RankSim}} \tag{5}$$

where γ is a hyperparameter that controls the magnitude of the W-RankSim regularizer. W-RankSim encourages weight vectors with closer class labels to be closer in cosine space. Furthermore, L_{main} ensures that feature vectors z_i are centered at their corresponding weighted vectors w_j. As z_i approaches the corresponding w_j, it simultaneously moves closer to other feature vectors with closer labels and moves away from those with farther labels.

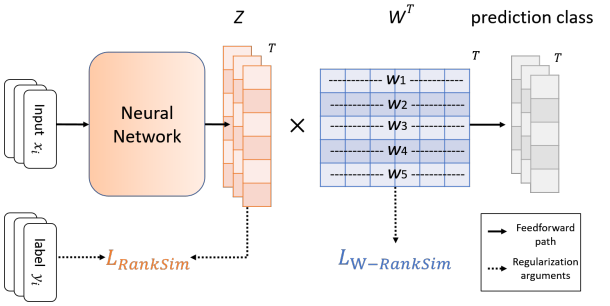

Figure 1: *Illustration of our proposed regularization W-RankSim vs. RankSim. RankSim requires the last feature embeddings and labels in a batch, whereas W-RankSim leverages weighted vectors in the output layer without being constrained on batch size to achieve a similar effect.*

In optimization, the **rk** function in Eq. 4 is piece-wise constant and non-differentiable. We adopt the method proposed in [20] to reformulate the **rk** function as a combinatorial objective:

$$\mathbf{rk}(a) = \arg\min_{\pi \in \prod_n} a \cdot \pi \tag{6}$$

where $\prod_n$ is a set containing all permutations of $\{1, 2, \ldots, n\}$. This formulation enables us to utilize blackbox combinatorial solvers [21] for differentiation.

The update formula is parameterized by a hyperparameter λ, which balances the faithfulness to the original function and the informativeness of the gradient, and is expressed by

$$\frac{\partial L}{\partial a} = -\frac{1}{\lambda}(\mathbf{rk}(a) - \mathbf{rk}(a_\lambda)),\ a_\lambda = a + \lambda \frac{\partial L}{\partial \mathbf{rk}} \tag{7}$$

For further details, please refer to [20, 21].

2.2. Comparing with RankSim

As illustrated in Figure 1, our proposed method directly imposes constraints on the weighted vectors, ensuring that ordinal information between each class is considered at each updating step, regardless of whether there is data imbalance. In contrast, RankSim is confined to the samples within the batch. Although it employs a subsampling technique to ensure that each label is considered only once in a batch, it still encounters inherent limitations, particularly in imbalanced ordinal classification tasks where the number of labels is typically small. In such cases, a large enough batch size is needed to ensure that the labels in a batch can offer sufficient variation and information. However, in our ASA task, a large batch size is not feasible due to the large input signal. In summary, W-RankSim is an approach that is not constrained by batch size and achieves, or even surpasses, the effectiveness of RankSim with lower overhead in ordinal classification tasks.

2.3. Hybrid model

Figure 2 illustrates our proposed hybrid model, which integrates SSL features with handcrafted features. The model consists of three main components: (a) content, (b) delivery, and (c) language use.

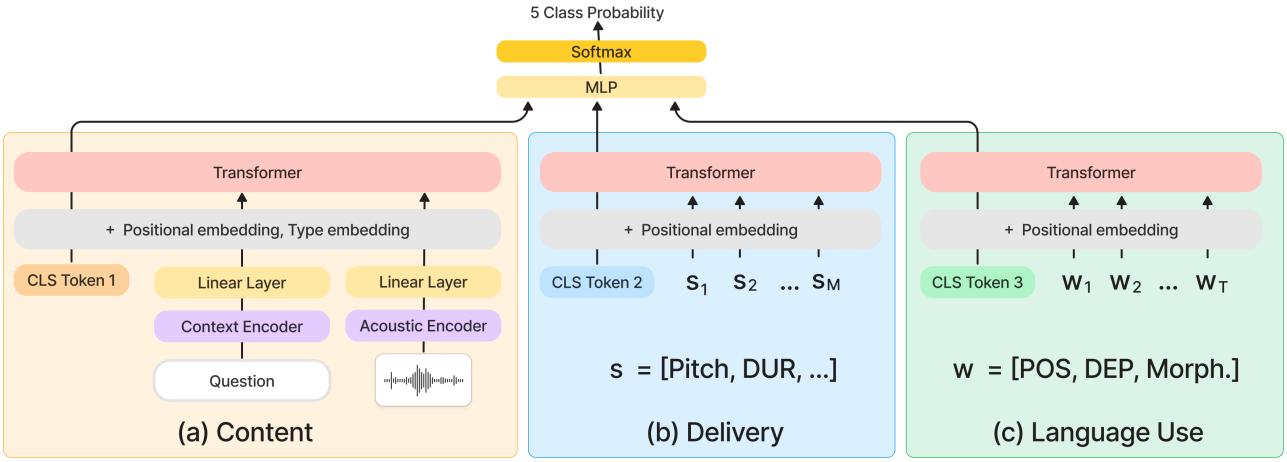

Figure 2: *The architecture of the Hybrid ASA model comprises three parts: (a) content, (b) delivery, and (c) language use, each addressing specific aspects of speech assessment. The first part utilizes SSL features generated by a pretrained acoustic model such as Whisper. The remaining parts leverage hand-crafted features to capture relevant characteristics for speech assessment.*

2.3.1. *Content component*

This component aims to evaluate the relevance between the user response and the question. We begin by encoding the audio signal and question context separately using pretrained context encoder and acoustic encoder, specifically BERT and Whisper, to obtain sequence embeddings that represent the questions and audio sequences while preserving contextual information. Subsequently, these embeddings are projected into the same space with H dimension using a linear layer. To distinguish between the embeddings of the two modalities (text and audio) and to capture the temporal sequence characteristics, we introduce learnable type embeddings and positional embeddings. Finally, we employ the CLS token by concatenating it with all embeddings. This concatenated representation is then passed through bidirectional transformer encoder [22] blocks to learn vector representations of the content features.

2.3.2. *Delivery component*

This aspect measures users' stress, pronunciation, fluency, etc. In this component, we utilize monolingual wav2vec2 2.0 for force alignment with ASR transcriptions, recognized by multilingual wav2vec 2.0, to segment the audio signal into word-level segments using the CTC segmentation algorithm [23]. Within each segmentation, we extract various features including pitch, duration (DUR), intensity, following silence, posterior probability from both monolingual and multilingual wav2vec 2.0, LM score from n-gram LM during transcript, and confidence score from multilingual wav2vec 2.0. These features are utilized to construct segment features s_i, and subsequently, a sequence of segment features $[s_1, s_2, \ldots, s_M]$ is concatenated to represent the delivery feature, where M is a predefined length achieved by either truncating or padding with zeros. To learn vector representations of these delivery features, we employ the same transformer encoder blocks, learnable positional embeddings, and CLS token as utilized in the content component.

2.3.3. *Language use component*

Language use measures users' abilities in grammar and syntax. In this component, we obtain a word sequence from the ASR transcription generated by the delivery component and adjust it to length T by either truncating or padding with zeros. For each word, we utilize spaCy, an industrial-strength natural language processing toolkit in Python, to extract Part-Of-Speech (POS), dependency labels (DEP), and morphology (Morph.). These features are then used to construct a language use embedding w_i of a word using one-hot encoding. Subsequently, a sequence of embeddings is concatenated $[w_1, w_2, \ldots, w_T]$ to represent the sequence. Finally, we adapt the same steps as in the delivery component, using the CLS token to represent the entire sequence.

These three components form the foundation for comprehensive evaluation [4, 15]. Similar to the approach proposed in [18], we incorporate prompting to adapt to various questions. Finally, the CLS token embeddings obtained from these three components are concatenated and passed through a multilayer perceptron (MLP) with H^o hidden dimensions. We apply the softmax function to the output to obtain the probability distribution over $|C|$ classes. In our case, $|C|$ is 5.

3. Experiments and Results

3.1. Data

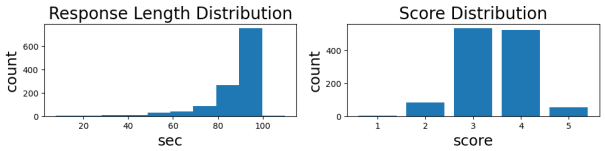

Figure 3: *The left figure shows response length distribution in the GEPT corpus, while the other displays score distribution.*

The corpus used in this study was privately collected by the Language Training and Testing Center from the General English Proficiency Test (GEPT) intermediate level exam, specifically focusing on the picture description module. This exam is a high-stakes English assessment test. The corpus comprises 1199 responses evenly distributed across 4 sets of questions, each provided by a different test taker. Participants were instructed as follows: "Below are a picture and four related questions. Please complete your answers in one and a half minutes. Do not read the number or the question when you answer. Please first look at the picture and think about the questions for thirty seconds."

The score distribution and response length distribution are illustrated in Figure 3. Scores were calculated as averages from

assessments provided by two professionals, ranging from 1 to 5, and any floating-point values were discarded unconditionally.

We randomly selected a specific set of questions to create an unknown content test set. This set was designed to assess performance under cold start conditions [18], where the model encounters previously unseen content. The remaining data was divided into train, development, and known content test sets in an 8:1:1 ratio. This partitioning strategy ensures that the model is trained on a majority of the data while allowing for robust evaluation on both familiar and novel content.

3.2. Experimental Setup

We employed Whisper-base as our acoustic encoder and utilized Sentence-BERT [24], specifically the all-MiniLM-L6-v2 model, as our context encoder. The dimensions for both the hidden layers (H) and the hidden layers in the MLP (H^o) were set to 512 and 64, respectively. During training, we utilized the RAdam optimizer [25] with a weighted decay of 1e-5. The learning rate was set to 2e-4, and training was conducted for 8 epochs with a batch size of 2.

The experiments were conducted with W-RankSim and RankSim using both cross-entropy loss and large margin cosine loss (LMCL) [26]. We set the hyperparameters λ and γ to 2 and 1.5, respectively, for W-RankSim and RankSim. LMCL has two hyperparameters, s and m, which were set as 1.96 and 0.15 in all experiments. Further details about s and m can be found in [26]. This configuration allowed us to effectively train our model while optimizing performance.

3.3. Performance comparison

Hybrid model	Known Content Accuracy	Unknown Content Accuracy
cross entropy loss	0.689	0.620
+W-RankSim	0.678	0.700
LMCL	0.678	0.687
+W-RankSim	**0.700**	**0.720**
w/o language use component	**Known Content Accuracy**	**Unknown Content Accuracy**
cross entropy loss	0.633	0.683
+W-RankSim	0.644	0.700
LMCL	0.644	0.700
+W-RankSim	**0.656**	**0.717**
w/o delivery and language use component	**Known Content Accuracy**	**Unknown Content Accuracy**
cross entropy loss	0.633	0.657
+W-RankSim	0.644	**0.697**
LMCL	0.644	0.680
+W-RankSim	**0.678**	0.683

Table 1: *Experiment results on the GEPT corpus. "Known Content Accuracy" denotes the accuracy on the known content test set, while "Unknown Content Accuracy" represents the accuracy on the unknown content test set.*

In Table 1, we conduct an ablation study on model components and demonstrate the effectiveness of W-RankSim in each model. We also experimented with various loss functions, including cross-entropy loss and LMCL. Across each model and loss function, incorporating W-RankSim consistently improved performance, indicating its benefit in ordinal classification tasks with imbalanced data.

The experiments demonstrate that both cross-entropy loss and LMCL combined with W-RankSim have improved performance in known content test set and unknown content test set, particularly with the inclusion of LMCL. LMCL in conjunction with W-RankSim consistently yields better performance in most models compared to using cross-entropy with W-RankSim. According to our observations, LMCL enhances local relations in cosine space, while W-Ranksim emphasizes global relations between each class in cosine space, thus improving performance. This suggests that leveraging these methods together enhances the classification accuracy.

Our baseline model consists solely of the content component. The experiments presented in Table 1 demonstrate the advantages of integrating handcrafted features. Both language use and delivery features contribute to performance improvement, with the fully hybrid model achieving the best performance on both the known content test set and the unknown content test set

3.4. Experiments on varying batch size.

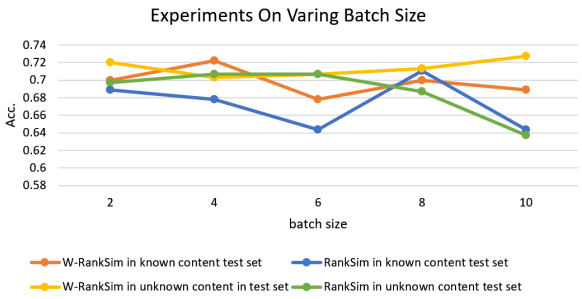

Figure 4: *Experiments tested a hybrid model that combined LMCL with different regularization across different batch sizes.*

We conducted experiments varying the batch size using the hybrid model with LMCL combined with W-RankSim and RankSim to assess the sensitivity of batch size in W-RankSim and RankSim. The results depicted in Figure 4 demonstrate that performance in W-RankSim is more consistent compared to RankSim when changing the batch size, and W-RankSim can achieve the best or near-best performance at lower batch size.

4. Conclusions

We approached the ASA task as an ordinal classification task with imbalanced data and introduced a novel regularization method, W-RankSim, specifically designed for this task. W-RankSim captures ordinal information between weighted vectors, indirectly encouraging embeddings to learn proximity and distance relations in both label and feature space. Our experiments consistently demonstrate that W-RankSim can improve performance across various models. Subsequently, we proposed a hybrid model that combines SSL features with traditional hand-crafted features and demonstrated its effectiveness through extensive experiments. Finally, the hybrid model using LMCL with W-RankSim achieved the best accuracy and exhibited steady performance across varying batch sizes. This underscores the robustness and efficacy of the proposed approach.

Limitations and future work. In this paper, we focus on exploring regularization techniques and feature extraction methods to enhance ASA systems. However, we acknowledge the significance of linguistic and phonetic aspects in real-world speech assessment, which were not fully addressed in our work. Future effort will involve collaborating closely with linguistics and phonetics experts to develop a more comprehensive ASA system that integrates these crucial factors. Additionally, we aim to investigate the applicability of W-RankSim in other ordinal classification tasks to advance classification techniques further and explore its potential in broader contexts.

5. Acknowledgement

This work was supported by the Language Training and Testing Center (LTTC), Taiwan. Any findings and implications in the paper do not necessarily reflect those of the sponsor.

6. References

[1] M. Eskenazi, "An overview of spoken language technology for education," *Speech communication*, vol. 51, no. 10, pp. 832–844, 2009.

[2] B. Lin and L. Wang, "Attention-based multi-encoder automatic pronunciation assessment," in *Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 7743–7747.

[3] K. Zechner, D. Higgins, X. Xi, and D. M. Williamson, "Automatic scoring of non-native spontaneous speech in tests of spoken English," *Speech Communication*, vol. 51, no. 10, pp. 883–895, 2009.

[4] Y. Qian, P. Lange, K. Evanini, R. Pugh, R. Ubale, M. Mulholland, and X. Wang, "Neural approaches to automated speech scoring of monologue and dialogue responses," in *Proceedings of the 2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2019, pp. 8112–8116.

[5] Z. Yu, V. Ramanarayanan, D. Suendermann-Oeft, X. Wang, K. Zechner, L. Chen, J. Tao, A. Ivanou, and Y. Qian, "Using bidirectional lstm recurrent neural networks to learn high-level abstractions of sequential features for automated scoring of non-native spontaneous speech," in *Proceedings of the 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 338–345.

[6] L. Chen, J. Tao, S. Ghaffarzadegan, and Y. Qian, "End-to-end neural network based automated speech scoring," in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 6234–6238.

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

[8] W.-N. Hsu, A. Sriram, A. Baevski, T. Likhomanenko, Q. Xu, V. Pratap, J. Kahn, A. Lee, R. Collobert, G. Synnaeve *et al.*, "Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training," *arXiv preprint arXiv:2104.01027*, 2021.

[9] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin *et al.*, "Superb: Speech processing universal performance benchmark," *arXiv preprint arXiv:2105.01051*, 2021.

[10] H. S. Bovbjerg and Z.-H. Tan, "Improving label-deficient keyword spotting through self-supervised pretraining," in *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSP)*, 2023, pp. 1–5.

[11] X. Xu, Y. Kang, S. Cao, B. Lin, and L. Ma, "Explore wav2vec 2.0 for mispronunciation detection." in *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2021, pp. 4428–4432.

[12] S. Siriwardhana, T. Kaluarachchi, M. Billinghurst, and S. Nanayakkara, "Multimodal emotion recognition with transformer-based self supervised feature fusion," *IEEE Access*, vol. 8, pp. 176 274–176 285, 2020.

[13] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proceedings of International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.

[14] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[15] S. Park and R. Ubale, "Multitask learning model with text and speech representation for fine-grained speech scoring," in *Proceedings of the 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023, pp. 1–7.

[16] E. Kim, J.-J. Jeon, H. Seo, and H. Kim, "Automatic pronunciation assessment using self-supervised speech representation learning," *arXiv preprint arXiv:2204.03863*, 2022.

[17] S. Bannò and M. Matassoni, "Proficiency assessment of L2 spoken english using wav2vec 2.0," in *Proceedings of the 2022 IEEE Spoken Language Technology Workshop (SLT)*, 2023, pp. 1088–1095.

[18] J. Park and S. Choi, "Addressing cold start problem for end-to-end automatic speech scoring," *arXiv preprint arXiv:2306.14310*, 2023.

[19] Y. Gong, G. Mori, and F. Tung, "Ranksim: Ranking similarity regularization for deep imbalanced regression," *arXiv preprint arXiv:2205.15236*, 2022.

[20] M. Rolínek, V. Musil, A. Paulus, M. Vlastelica, C. Michaelis, and G. Martius, "Optimizing rank-based metrics with blackbox differentiation," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7620–7630.

[21] M. V. Pogančić, A. Paulus, V. Musil, G. Martius, and M. Rolinek, "Differentiation of blackbox combinatorial solvers," in *Proceedings of the International Conference on Learning Representations*, 2019.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[23] L. Kürzinger, D. Winkelbauer, L. Li, T. Watzel, and G. Rigoll, "CTC-segmentation of large corpora for German end-to-end speech recognition," in *Proceedings of the 2020 International Conference on Speech and Computer (SPECOM)*, 2020, pp. 267–278.

[24] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[25] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," *arXiv preprint arXiv:1908.03265*, 2019.

[26] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5265–5274.