Interspeech 2024
1-5 September 2024, Kos, Greece

The Whole Is Bigger Than the Sum of Its Parts: Modeling Individual Annotators to Capture Emotional Variability

James Tavernor[1], *Yara El-Tawil*[1], *Emily Mower Provost*[1]

[1]University of Michigan, USA

tavernor@umich.edu, yeltawil@umich.edu, emilykmp@umich.edu

Abstract

Emotion expression and perception are nuanced, complex, and highly subjective processes. When multiple annotators label emotional data, the resulting labels contain high variability. Most speech emotion recognition tasks address this by averaging annotator labels as ground truth. However, this process omits the nuance of emotion and inter-annotator variability, which are important signals to capture. Previous work has attempted to learn distributions to capture emotion variability, but these methods also lose information about the individual annotators. We address these limitations by learning to predict individual annotators and by introducing a novel method to create distributions from continuous model outputs that permit the learning of emotion distributions during model training. We show that this combined approach can result in emotion distributions that are more accurate than those seen in prior work, in both within- and cross-corpus settings.

Index Terms: speech recognition, emotion recognition, human-computer interaction, inter-annotator agreement

1. Introduction

Expressions of emotion are nuanced and complex, and people perceive these expressions differently, adding to the complexity. Most emotion recognition models overlook this nuance [1]. This is because most Speech Emotion Recognition (SER) datasets and tasks present the ground truth as a single label, which is the average of multiple annotations. In this work, we present novel approaches to both accurately learn the perceptions of individual annotators and aggregate these estimates to create distributions of annotator perception. In this way, the model retains information about individual annotator predictions while still being able to summarize the information accurately as a two-dimensional (2D) distribution.

Prior work has investigated methods to retain information about variability and uncertainty. Research has included the prediction of measures such as unbiased annotator standard deviation [2, 3], the embedding of individual annotators to improve performance on the aggregated ground truth, with some investigation into how well the model annotator uncertainty correlates with real uncertainty [4, 5, 6], and the prediction of the distribution of annotations over a given utterance [7]. Yet, gaps remain. Methods that summarize model information or predict uncertainty lose fine-grained information about individual annotators. On the other hand, methods that seek to learn annotators primarily do so to improve performance on the aggregated ground truth or investigate much smaller numbers of annotators than are generally used in these datasets.

We present a novel approach that predicts the annotations of individuals and includes a new differentiable method to automatically learn distributions similar to [7], enabling the modeling of individual variation and the retention of the ability to summarize annotators. The model training involves an interleaved approach, alternating between different tasks: learning individual annotators and learning a distribution. We learn individual annotators by training a multi-task (MT) model to predict each annotator in the training set across the dimensions of valence and activation. We learn a distribution by upsampling the observations from the MT model and using Kernel Density Estimation (KDE) to produce a summarization of the model output as a distribution. We introduce differentiable KDE into the model training process to enable the use of gradient descent.

We present both within- and cross-corpus investigations. Within-corpus, we find that a model trained with the interleaved tasks of individual annotator perception and distribution learning can outperform a method that learns to predict the distribution alone [7], in terms of both the performance on consensus labels and the accuracy of the distribution itself, while providing individual annotations as well. We further show that the output of the annotator-specific models (trained only on annotator prediction) can be post-processed to create a distribution, rather than learning a distribution during model training, that outperforms the prior work of [7]. In this case, an extra step is involved in which the output of the annotator-specific models is transformed into a distribution using either KDE as in [7] or using the differentiable KDE method presented in this work. We find that using differentiable KDE leads to significantly improved performance, even when only used in post-processing, pointing to the efficacy of this approach for either model learning or post-hoc output summarization. Cross-corpus, we demonstrate that annotator-specific models can be used zero-shot without knowledge about the annotators that labeled the new datasets. We find that the presented approach outperforms a distribution-only method across metrics that capture individual annotators and the accuracy of a given distribution in most cases. Future work will focus on investigating individual characteristics of annotators (e.g., personality) and how this information can also be considered when learning annotator-specific perception.

2. Related Work

Previous work has developed soft-label methods that use multiple annotators per label. Dang et al. [8] use multi-rater Gaussian Mixture Regression to make temporal emotion predictions for a fixed set of consistent evaluators in their target dataset. Other approaches have captured both the uncertainty in annotator labels and model uncertainty [9]. However, a gap remains at the intersection of predicting individual annotations for a variable number of annotators.

Instead, we build on the label processing method devel-

10.21437/Interspeech.2024-2307

oped in previous work by Zhang et al. [7], which incorporated inter-annotator variance into machine learning models by creating new ground truth labels that incorporate this knowledge [7]. They upsampled existing annotations by selecting random subsets of annotators for each utterance and took the mean across those annotations. They added random noise to the resulting means, such that $x\ noise \sim U(-\frac{std(x)}{2}, \frac{std(x)}{2})^1$, where x is activation or valence, std indicates the standard deviation of the annotator ratings for that utterance, and U is the uniform distribution. Kernel Density Estimation (KDE) via Diffusion was then calculated over the upsampled observations. They divided the KDE output grid into N bins for each dimension and took the mean over the KDE samples inside each bin. They converted this grid to a probability distribution by normalizing over the means. The authors investigated $N = 2$ and $N = 4$. The KDE step was essential to remove sensitivity to where boundaries were drawn. The authors then trained a model to predict these binned distributions. However, in this approach, the model loses information about individual annotators. Additionally, because the approach is not differentiable, it cannot be included in model training. We present an approach with a differentiable component that permits learning a binned distribution, implemented using sigmoid-based soft operations.

Previous work has investigated the prediction of individual annotators on subjective tasks such as emotion recognition and hate speech [4, 5, 10]. Davani et al. introduced an encoder-based model with separate classification heads for each annotator. They trained this model for a binary categorical text emotion recognition task using a dataset that contained 82 annotators. At test time, they aggregated the individual annotator predictions and found that their model outperformed a baseline trained on majority ground truth labels. However, the performance of individual annotators was not discussed. Further, a limitation of this work is that many SER datasets include over 82 annotators, and the authors acknowledge that it would be too computationally expensive to train a model with separate heads for large numbers of annotators. Previous work has shown that clustering similar annotators can mitigate problems with large numbers of annotators [11]. However, clustering annotators loses information about individual ratings. In our work, we enable only the relevant heads per batch, making training with a large number of annotators more computationally feasible.

An alternative approach to learning individual annotators is through annotator embeddings [5]. Prior work from Kocoń et al. demonstrates that annotator-specific embeddings can be used to personalize model predictions and capture the bias of individual annotators. They introduced four methods for encoding annotator information into the model, including a one-hot annotator embedding. This embedding was a one-hot encoded vector of annotator ID that was concatenated to the model input. They found that this led to improved text-based emotion predictions but were focused on a consensus model rather than an individual-specific model. We use the one-hot model and investigate if the model can learn individual annotators.

3. Experiments

3.1. Data setup

We use the MSP-Improv dataset for training and testing. It was labeled using crowdsourcing and has a relatively large number of evaluations per utterance [7, 12]. Additionally, we use the IEMOCAP, MSP-Podcast, and MuSE datasets to evaluate the cross-corpus results of each method.

MSP-Improv is an SER dataset consisting of acted improvised dialogue designed to evoke certain emotions [12]. The dataset has 12 speakers evenly split between male and female actors across six sessions. We select a speaker-independent data split such that all annotators in the validation and test set have evaluated at least one utterance in the training set. Annotators will be present in the training set that do not appear in the validation or test set (for example, when the annotator annotated less than three samples). The resulting train, validation, and test split size is 5,851, 1,287, and 1,300 utterances, respectively[2]. The training set was evaluated by 1,434 individual crowdsourced annotators, with each sample receiving between 5 and 50 annotations (mean of 7.2). A subset of these annotators evaluated the validation and test set (1,305 and 1,197, respectively). Both validation and test set samples have between 5 and 37 evaluations per sample, with a mean of 7.3 and 7.6 annotators. In few samples (28) the same annotator has annotated more than once. In these cases we have averaged their annotations into one evaluation, and adjusted the mean ground truth for these samples.

The **IEMOCAP** dataset contains five dialogue sessions containing scripted and improvised interactions between two actors. There is one female and one male actor in each conversation [13]. We remove utterances where individual annotations were partially missing or any annotator evaluations were not within the labeling range described in the data collection. After processing, the dataset consists of 9,999 samples. Six annotators labeled the dataset with an average of 2.13 annotators per sample. We test on the full dataset.

MSP-Podcast is a dataset of speech taken from podcasts and then labeled [14]. We use the predefined splits and evaluate on test_set_1, which is comprised of 13,911 utterances and contains 9570 individual annotators. Each utterance was evaluated by 6.9 crowdsourced annotators on average. We use `release 1.8`, which does not contain transcripts, so we use Microsoft Azure automatic speech recognition to generate them.

MuSE is a dataset of 28 college students recorded in two 45-minute sessions each, responding to emotional stimuli. One session was when the students were affected by an external stressor, and the other was without the stressor [15]. Students were recorded using a lapel microphone. Crowdsourced annotators evaluated each utterance. There are 2,584 utterances comprised of 1,385 stressed and 1,199 non-stressed samples. The dataset provides labels annotated with or without context; we use the labels from the 160 individual annotators who labeled without context. Each sample was evaluated between 7 and 9 annotators, with 8 on average.

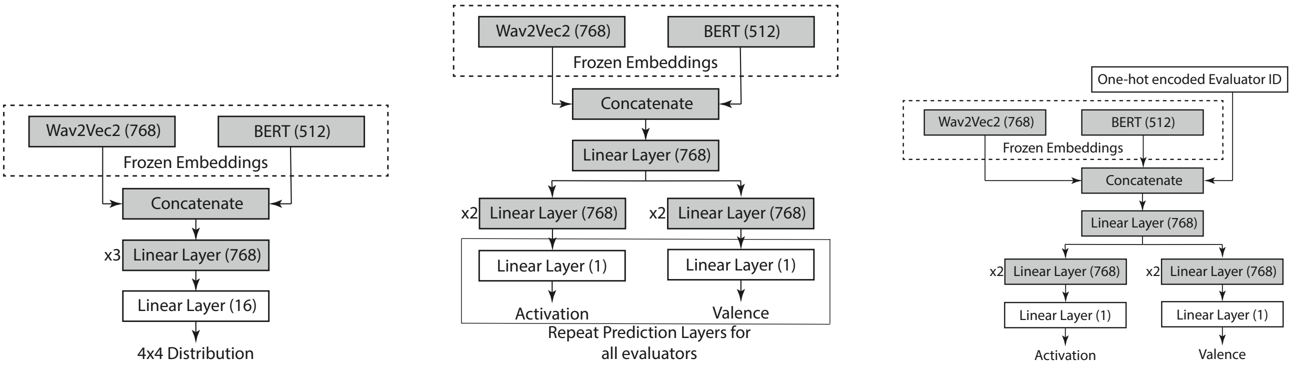
Dataset Preprocessing We process all datasets in the same way. We use min-max scaling on the annotator and consensus labels for activation and valence to restrict labels to the $[-1, 1]$ range. We then use KDE to generate a 2D ground truth probability distributions as in [7]. We use a KDE grid size of 512 as we assume this will be sufficiently large to ensure the probability is insensitive to the grid boundaries.

3.2. Model Architecture

We present three models: a baseline, a MT model, and a one-hot model, all of which share the same base architecture but have different output head architectures (Figure 1). The model input includes the frozen mean-pooled final layers of Wav2Vec2 [16]

[1] We also add $\epsilon = 1E^{-12}$ to this value to account for cases where standard deviation is 0.

[2] The code to create the data splits can be found at https://github.com/chailab-umich/ModelingIndividualEvaluators.



(a) *Baseline model*

(b) *MT Model*

(c) *One-hot Model*

Figure 1: *Model Architectures. Layers in gray are the common architecture between models. In (b) and (c) the last two common layers are duplicated as they split the model to two predictions.*

and BERT [17] CLS embeddings as these have shown effectiveness in SER applications [18, 19]. We apply dropout with probability 0.2 and concatenate the embeddings. The concatenated embedding is passed through a single linear layer of size 256 with ReLU activation. For each prediction (distribution, activation, or valence), the input will pass through two linear layers of size 256 with ReLU activations.

The **baseline** model directly learns the generated KDE distribution (as in [7]), having a final linear layer output of 16 logits for the 4x4 discretized KDE distribution prediction case. The **MT** model has separate prediction layers for each annotator as in [17]. Each annotator's continuous prediction of activation and valence is made via a linear layer with an output size of 1. We use the **one-hot** method, previously used for text emotion recognition [20]. We use the same architecture as in the MT case but with only one annotator prediction head. The annotator ID is one-hot encoded and concatenated to the Wav2Vec2 and BERT embeddings on the model input. When training within corpus, we reduce computation cost by making predictions only for annotators in the batch input to the model.

3.3. Training Tasks

In this section, we define three different training tasks. The Baseline model is trained with the Baseline task. MT and one-hot models are trained by interleaving Tasks 1 and 2 (Task 1+2), defined below. We use stochastic gradient descent with a learning rate of 0.001, with a learning rate scheduler that adjusts the learning rate by a factor of 0.1 after five epochs of no reduction in validation metrics. We train models until early stopping triggers with a patience of 10 with a minimum of 30 epochs. Each model trains with a batch size of 32. For all methods we use the relevant task's validation losses.

Baseline: We predict the flattened 2D distribution and use cross-entropy loss of the 16 logits output against the flattened 2D generated ground truth probability distributions [7].

Task 1 - Annotator Training: We train annotator-specific predictions using the individual annotator ground truth. We use Lin's Concordance Correlation Coefficient (CCC) loss as our loss function since it better models dimensional attributes than other regression losses [21]. The sets act, and val contain the ground truth labels from all annotators in a training batch. The sets m_{act} and m_{val} contain the model's estimates of these labels. The loss is $2 - CCC(m_{act}, act) - CCC(m_{val}, val)$.

Task 2 - *DiffKDE*: We learn the probability distribution using the KDE-generated ground truth labels. The model must produce a probability distribution from the model's activation and valence predictions. However, the KDE method outlined by Zhang et al. in [7] is not immediately usable. KDE via diffusion starts with a histogram [22]. For each annotation we must know if it is in a particular bin to increment the bin's histogram count. This operation is a binary operation and not differentiable. We introduce a differentiable approximation to this problem by instead calculating a confidence value that a given annotation is within a given bin. We modify an existing one-dimensional (1D) soft-histogram[3], as below, for the 2D data.

We use 64 bins for *DiffKDE*[4]. We first calculate the 1D center of each bin in the range -1 to 1. For each of the n annotations of activation, we subtract the center of each bin from the annotation, resulting in a 64 size vector, which we call x. The contribution to the 64 bins will then be calculated using an element-wise $sigmoid$ on this vector, $sigmoid(\sigma * (x + \frac{\delta}{2})) - sigmoid(\sigma * (x - \frac{\delta}{2}))$. The gradient of $sigmoid$ is largest at zero, for values of x far from 0, the $\frac{\delta}{2}$ term has less effect, and the bin value is close to 0. This function is maximized for values of x close to 0. We repeat this for valence to get two $n \times 64$ matrices for activation and valence.

In the equation, σ is a scaling parameter; the larger the value, the more sharp the histogram is, and δ is the bin size. Since our data is in the range -1 to 1, and we use 64 bins, $\delta = \frac{2}{64}$. We then matrix multiply these two $n \times 64$ matrices by transposing one to get a 2D (64×64) matrix. We then normalize to get a final 4×4 probability distribution as in [7]. There is a tradeoff where too large of a σ may lead to vanishing gradients, but too low may result in undersaturation [23]. As such, we set σ relatively small at 8; lower values did not reduce loss. Future work could investigate the impact of modifying the σ parameter. The generation of probabilities in *DiffKDE* is done in `float16`[5] as it significantly speeds up calculations.

We base our work off an existing KDE via Diffusion library[6], which we modify to use PyTorch and the soft histogram method from the previous paragraph. All code is available on our GitHub page[7]. This enables *DiffKDE* to be run on GPUs and parallelized into batches. *DiffKDE* Loss is the Cross-Entropy loss[8] of the *DiffKDE* output, compared with the generated ground-truth 2D labels.

[3] https://discuss.pytorch.org/t/differentiable-torch-histc/25865/4

[4] Note: smaller than 512 (used to generate target labels) for speed

[5] We use `float64` during validation and testing for KDE accuracy.

[6] https://pypi.org/project/KDE-diffusion/

[7] https://github.com/chailab-umich/ModelingIndividualEvaluators

[8] After normalization, we add $\epsilon = 1E^{-8}$ to avoid taking log of 0.

Table 1: *MSP-Improv probability distribution results (*=statistical significant improvement compared to baseline, †=statistical significant decline). ↑ indicates higher is better, ↓ indicates lower is better. Each metric's best result is bolded.*

Model	TVD↓	JSD↓	Activation CCC↑	Valence CCC↑
Baseline	.515±.004	.213±.003	.673±.008	.573±.020
MT	**.503±.001***	**.211±.001**	**.741±.005***	.571±.005
One-hot	.518±.006	.228±.004†	.689±.014*	**.607±.017***

3.4. Evaluation Metrics

We first evaluate the ability of the proposed approaches to learn continuous predictions and then the ability of the system to learn distributions. The baseline cannot directly produce continuous ratings, while the proposed approaches can. In order to provide a fair comparison, we generate consensus predictions across all methods in the same manner: we sum along the activation/valence dimensions and then multiply this sum with $[-1, -0.5, 0.5, 1]$. We use CCC to measure the systems' ability to predict individual annotators' labels (note: we cannot evaluate the baseline for this task). Next, we evaluate the consensus predictions by comparing them to the averaged ground truth using CCC. Finally, we measure the differences between the learned and ground truth probability distributions using Total Variation Distance (TVD), and Jensen-Shannon Divergence (JSD) [7][9]. Test results are reported over five seeds. Significance asserted at a 5% confidence on a paired two-sided t-test.

4. Results

4.1. MSP-Improv Results

The MT approach predicts annotator-specific activation more accurately than the one-hot model (0.629 ± 0.002 vs. 0.349 ± 0.015, respectively) while the one-hot model has stronger performance for valence (0.393 ± 0.006 vs. 0.429 ± 0.007, respectively). The consensus output for both the MT and one-hot approaches show significant improvements in activation CCC compared to the baseline. In contrast, only the one-hot method significantly improves valence. Overall, we find that the MT model learns more accurate distributions compared to the baseline when using the soft-histogram across both metrics, showing signficant improvement over the baseline for TVD. The one-hot method has comparable TVD and statistically significantly worse JSD than the baseline. See Table 1 for more details.

4.2. Ablation Results

We investigate the importance of the interleaved training tasks for learning the distributions. In the previous experiments, we used *DiffKDE* during training and testing (Task 1+2). When using Task 1 alone, no distribution is used during training, so the best method to build the distribution is uncertain. We generated results for Task 1 alone using both *KDE* and *DiffKDE* to generate distributions. We find that when using *DiffKDE* there is a significant performance increase for TVD (0.553±0.003 to 0.500±0.003) and JSD (0.265±0.002 to 0.211±0.002). This is very similar to the performance of Task 1+2 (Table 1).

4.3. Cross-Corpus Results

In a cross-corpus (zero-shot) context, the model does not have information about all annotators in advance. Therefore, we use all annotator predictions from the model. We use the MT approach as it has generally outperformed one-hot models. The MT models excel in cross-corpus performance and significantly outperform the baseline in all probability distribution measures (TVD and JSD) on all datasets. Additionally, we find statistically significant increases in Activation CCC performance on the IEMOCAP and MuSE datasets for both the annotator-only trained model (Task 1) and the interleaved tasks trained model (Task 1+2). The outlier is Valence CCC, which generally decreases compared to the baseline. See Table 2.

Table 2: *Cross-corpus zero-shot* ***Act****ivation,* ***Val****ence results, *,†,↑,↓ as in Table 1. P: MSP-Podcast, I: IEMOCAP, M: MuSE*

	Dataset	Baseline	MT-1	MT-12
TVD↓	P	0.601±0.003	**0.507±0.002***	0.518±0.005*
	I	0.633±0.002	0.614±0.002*	**0.613±0.002***
	M	0.530±0.004	0.484±0.007*	**0.470±0.002***
JSD↓	P	0.274±0.002	**0.213±0.001***	0.220±0.003*
	I	0.310±0.002	**0.302±0.002***	**0.302±0.002***
	M	0.218±0.003	0.192±0.005*	**0.182±0.002***
Act. CCC↑	P	**0.261±0.014**	**0.261±0.008**	0.235±0.012†
	I	0.374±0.010	**0.429±0.010***	0.381±0.015
	M	0.173±0.022	0.202±0.014	**0.209±0.012***
Val. CCC↑	P	**0.368±0.003**	0.332±0.009†	0.302±0.011†
	I	**0.321±0.011**	0.255±0.007†	0.219±0.011†
	M	0.198±0.017	**0.202±0.013**	0.162±0.007†

The MT models generally struggled with the valence dimension, showing significant decreases compared to the baseline. Given that we are using all annotators for zero-shot test time, it is likely many annotator predictions that did not learn valence well have influenced the valence dimension negatively. Ultimately, we believe that using all annotators as we have done in a zero-shot setting is not an upper bound for performance of these models. Instead, selecting a subset of trained annotators may significantly increase performance in the zero-shot setting.

5. Conclusion

Learning individual annotators is challenging. The model must learn a very large number of annotators across both the dimensions of activation and valence. We have presented an approach that accurately predicts individual annotators and a differentiable KDE operation that can be applied to a multi-task annotator models to produce distributions more accurately than using KDE to generate the distributions. We find that a multi-task model sufficiently learns the individual annotators to produce a probability distribution that outperforms methods that only learn distributions while retaining information about individual annotators. Furthermore, we have found significant improvement in multiple zero-shot settings when using the multi-task model over the baseline. We believe these methods can potentially increase utility to the end-user by providing more information about model predictions retained in the model. Future work also includes improving the capability of the model to capture valence, which will likely improve the distribution performance as well. Additionally, we believe the method provides avenues into studying how emotion models can predict specific to groups of annotators or leverage the knowledge of annotators to improve zero-shot cross-corpus performance.

[9] We use natural logarithm for JSD instead of $\log_2$

6. Acknowledgements

This material is based in part upon work supported by the National Science Foundation (NSF IIS-RI 2230172 and IIS-RI 2230172) and National Institutes of Health (NIH R01MH130411).

7. References

[1] S. Labat, N. Ackaert, T. Demeester, and V. Hoste, "Variation in the expression and annotation of emotions: a wizard of oz pilot study," in *LREC 2022 Workshop: 1st Workshop on Perspectivist Approaches to NLP (NLPerspectives)*. European Language Resources Association (ELRA), 2022, pp. 66–72.

[2] J. Han, Z. Zhang, M. Schmitt, M. Pantic, and B. Schuller, "From hard to soft: Towards more human-like emotion recognition by modelling the perception uncertainty," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 890–897.

[3] N. R. Prabhu, G. Carbajal, N. Lehmann-Willenbrock, and T. Gerkmann, "End-to-end label uncertainty modeling for speech-based arousal recognition using bayesian neural networks," *arXiv preprint arXiv:2110.03299*, 2021.

[4] A. M. Davani, M. Díaz, and V. Prabhakaran, "Dealing with disagreements: Looking beyond the majority vote in subjective annotations," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 92–110, 2022.

[5] J. Kocoń, M. Gruza, J. Bielaniewicz, D. Grimling, K. Kanclerz, P. Miłkowski, and P. Kazienko, "Learning personal human biases and representations for subjective tasks in natural language processing," in *2021 IEEE International Conference on Data Mining (ICDM)*, 2021, pp. 1168–1173.

[6] H.-C. Chou and C.-C. Lee, "Learning to recognize per-rater's emotion perception using co-rater training strategy with soft and hard labels." in *INTERSPEECH*, 2020, pp. 4108–4112.

[7] B. Zhang, G. Essl, and E. Mower Provost, "Predicting the distribution of emotion perception: capturing inter-rater variability," in *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, 2017, pp. 51–59.

[8] T. Dang, V. Sethu, and E. Ambikairajah, "Dynamic multi-rater gaussian mixture regression incorporating temporal dependencies of emotion uncertainty using kalman filters," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4929–4933.

[9] W. Wu, C. Zhang, and P. Woodland, "Estimating the uncertainty in emotion attributes using deep evidential regression," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 15 681–15 695.

[10] L. Stappen, L. Schumann, A. Batliner, and B. W. Schuller, "Embracing and exploiting annotator emotional subjectivity: An affective rater ensemble model," in *2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, 2021, pp. 01–08.

[11] S. G. Upadhyay, W.-S. Chien, B.-H. Su, and C.-C. Lee, "Learning with rater-expanded label space to improve speech emotion recognition," *IEEE Transactions on Affective Computing*, pp. 1–15, 2024.

[12] C. Busso, S. Parthasarathy, A. Burmania, M. AbdelWahab, N. Sadoughi, and E. M. Provost, "Msp-improv: An acted corpus of dyadic interactions to study emotion perception," *IEEE Transactions on Affective Computing*, vol. 8, no. 1, pp. 67–80, 2017.

[13] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower Provost, S. Kim, J. Chang, S. Lee, and S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Language Resources and Evaluation*, vol. 42, pp. 335–359, 12 2008.

[14] R. Lotfian and C. Busso, "Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings," *IEEE Transactions on Affective Computing*, vol. 10, no. 4, pp. 471–483, October-December 2019.

[15] M. Jaiswal, C.-P. Bara, Y. Luo, M. Burzo, R. Mihalcea, and E. M. Provost, "MuSE: a multimodal dataset of stressed emotion," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, Eds. Marseille, France: European Language Resources Association, May 2020, pp. 1499–1510. [Online]. Available: https://aclanthology.org/2020.lrec-1.187

[16] 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.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *North American Chapter of the Association for Computational Linguistics*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:52967399

[18] J. Tavernor, M. Perez, and E. Mower Provost, "Episodic Memory For Domain-Adaptable, Robust Speech Emotion Recognition," in *Proc. INTERSPEECH 2023*, 2023, pp. 656–660.

[19] J. Wagner, A. Triantafyllopoulos, H. Wierstorf, M. Schmitt, F. Burkhardt, F. Eyben, and B. W. Schuller, "Dawn of the transformer era in speech emotion recognition: closing the valence gap," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[20] J. Kocoń, M. Gruza, J. Bielaniewicz, D. Grimling, K. Kanclerz, P. Miłkowski, and P. Kazienko, "Learning personal human biases and representations for subjective tasks in natural language processing," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 1168–1173.

[21] B. T. Atmaja and M. Akagi, "Evaluation of error-and correlation-based loss functions for multitask learning dimensional speech emotion recognition," in *Journal of Physics: Conference Series*, vol. 1896, no. 1. IOP Publishing, 2021, p. 012004.

[22] Z. I. Botev, J. F. Grotowski, and D. P. Kroese, "Kernel density estimation via diffusion," *The Annals of Statistics*, vol. 38, no. 5, pp. 2916 – 2957, 2010. [Online]. Available: https://doi.org/10.1214/10-AOS799

[23] H. H. Tan and K. H. Lim, "Vanishing gradient mitigation with deep learning neural network optimization," in *2019 7th International Conference on Smart Computing Communications (ICSCC)*, 2019, pp. 1–4.