Trustworthy Majority Voting for Labeling and Analyzing Multi-Annotator Text Sentiment Datasets

Fotis Avgoustidis   Paraskevi Bassia   Prof. Ioannis Pitas
Department of Informatics, Aristotle University of Thessaloniki
Email: pitas@csd.auth.gr

***Abstract*—A typical way to label datasets for Deep Neural Network (DNN) training and testing is through crowdsourcing. However, there is no assurance that crowd workers will adhere to the data labeling criteria, refrain from introducing personal bias, or from spamming random labels. In order to address this issue, we propose a graph-based technique to assess annotator trustworthiness and adjust their involvement in the labeling process. Our proposed method not only improves data labels accuracy, by considering the agreement between annotators and ranking them based on their labeling trustworthiness, but also aims to enhance DNN inference performance by providing more accurate training data labels. We examine the constraints of conventional multi-annotation label aggregation techniques and compare them to our approach. Lastly, we demonstrate that our proposed method remains robust to artificially injected noisy annotations, surpassing the performance of previous state-of-the-Art (sotA) work. The effectiveness of the proposed method is validated on an intrinsically subjective task, namely text sentiment analysis.**

I. Introduction and Related Work

Data label accuracy is important in Natural Language Processing (NLP) and supervised Machine Learning (ML) applications. Valid text data class labels are essential for training effective Deep Neural Network (DNN) models in a range of ML or NLP tasks, including text sentiment analysis, entity recognition, and text categorization [1]. High-quality labeled DNN training data ensure better-performing DNN models, as supervised learning relies heavily on training data quality [2], [3]. Traditionally, human annotators either provide accurate data labels or ensure label correctness, when labels have been obtained otherwise, such as using automated ML methods. However, in subjective text analysis tasks, such as fine-grained text sentiment analysis, even honest human annotators output can provide inaccurate or ambiguous text labels [4]. This is evident in many cases of multi-annotator text datasets, where labels of each single text entry have been produced by multiple annotators. In such cases, the implementation of robust label aggregation techniques can provide reliable ground-truth data labels.

Various approaches have been suggested to address the issue of combining data labels from multiple annotators, each offering different advantages and drawbacks:

Simple Majority Voting (SMV) is a widely used approach where the final aggregated label is the one that received the most annotator votes [5]. While simple and practical, SMV treats all annotators uniformly, without taking into account their trustworthiness. This could lead to biased results, i.e., when some annotators provide, deliberately or not, false data labels.

Weighted Majority Voting (WMV) attempts to mitigate SMV limitations by assigning equal initial weights to all data annotators that are subsequently updated based on their overall consensus with the majority derived labels [6]. WMV does not consider the pairwise agreement between annotators, hence missing patterns of noisy annotation behaviors that do not deviate from the majority, but negatively impact both the annotated label quality and DNN training results.

Dawid-Skene model is a probabilistic approach that estimates true labels and error rates of annotator labeling using the Expectation-Maximization (EM) algorithm [7]. This model assumes that annotators have different levels of labeling expertise and can make systematic labeling errors. By modeling these errors, the Dawid-Skene model can provide more accurate label estimates compared to majority voting.

Bayesian methods extend the Dawid-Skene model by incorporating previous knowledge regarding the data labels. Bayesian inference is used to constantly update the accurate label probability estimation as further data are collected [8]. Bayesian methods can manage uncertainty and integrate previous information, making them useful tools for label cleaning.

Multi-Annotator Comptence Estimation (MACE) is a generative model designed to estimate both the true data labels and annotator trustworthiness in multi-annotator data scenarios [9]. MACE assumes that each annotator either correctly identifies the true label or produces a label at random, when spamming. It improves the data label quality by allocating more weight to trustworthy annotators.

DNN Crowd Layer (CL) can be used to integrate multi-annotator modeling directly in the DNN architecture, allowing for an end-to-end DNN training while accounting for annotator biases and reliabilities [10]. This method assumes the use of a primary DNN (e.g. a Convolution Neural Network) complemented by the CL that comprises of different parameters to weight the labels provided by each annotator. These CL parameters are updated during DNN CL training towards identifying the trend of each annotator. CL produces multiple outputs, one for each annotator. Each of the outputs predicts how a specific annotator would label the data input.

Multi-Annotator Loss Modeling [11] uses multi-task learning [12] and DNN training loss-based label correction [13] to improve DNN prediction accuracy and remain robust to label noise. Multi-Annotator Loss Modeling effectively separates agreeing and disagreeing label annotations to improve DNN prediction performance in various annotation settings.

A comprehensive review of learning methods from crowdsourced noisy labels is presented in [14].

Although the above mentioned methods have improved DNN training on multi-annotator noisy datasets, there are still obstacles to achieve a high level of agreement among annotators and create accurate labels, especially in tasks involving subjective label assessments.

To overcome them, we suggest a novel graph-based method for annotator ranking based on their trustworthiness. To this end, an *Annotator Agreement Graph* (AAG) is created, whose nodes represent annotators. AAG edge weights represent the level of agreement of an annotator pair during the label process. Using AAG, we can determine annotator trustworthiness. Thus, *Labeling Agreement Score* (LAS) can be assigned to each annotator. Then, LAS is utilized in a weighted *Trustworthy-Majority Voting* (TMV) scheme to aggregate multi-annotator labels.

This novel TMV method proposed in this paper advances the state-of-the-art by reinforcing existing approaches and emphasizing the trustworthiness of individual annotators and the accuracy of their annotations. This approach is particularly valuable in tasks involving subjective label assessments, where achieving a high level of agreement among annotators has traditionally been challenging.

The structure of this paper is as follows: Section II presents the TMV methodology. Experimental results are presented in Section III and conclusions are drawn in Section IV.

II. TMV Methodology

The proposed TMV method utilizes the label agreement among annotators to calculate their trustworthiness and generate accurate aggregated labels for DNN training . Our graph-based annotator ranking system consists of three steps: a) construction of an Annotator Agreement Graph, b) calculation of a Label Aggregation Score for each annotator, c) the Trustworthy Majority Voting Scheme.

Consider a DNN training dataset comprising N data sample vectors $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$. In the case of NLP, each column vector $\mathbf{x}_n$, $n = 1, \ldots, N$ represents a text entry, e.g. one tweet. These samples are annotated by a set of M annotators, denoted as $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$. Each text label corresponds to one of L classes labels $\mathcal{C} = \{c_1, c_2, \ldots, c_L\}$, corresponding, for instance, to text sentiments. Each sample $\mathbf{x}_n$ is annotated by a subset $\mathcal{S}_n$ of at most M annotators $1 \leq |\mathcal{S}_n| \leq M$. Annotator A_m, $m = 1, \ldots, M$ can provide a label $y_{nm} \in \mathcal{C}$ for a sample $\mathbf{x}_n$.

The *Annotator Agreement Graph (AAG)*, $G = \{\mathcal{V}, \mathcal{E}\}$ is constructed as follows: Its node set $\mathcal{V}$ comprises the annotators ($|\mathcal{V}| = M$). The AAG edge set $\mathcal{E}$ contains entries that connect annotator pairs. An edge (m, m') is formed between annotators A_m and $A_{m'}$ if they have annotated at least a minimum number of T common data samples. This parameter threshold T requires fine-tuning for each different DNN training dataset. The weight $w_{mm'}$ of each edge (m, m') is equal to the Cohen Kappa Score [15] between the two annotators, which quantifies the level of agreement between annotators on a set of jointly annotated text samples.

Once the AAG graph is constructed, for each annotator A_m, an average *Label Agreement Score (LAS)* a_m is calculated as follows:

$$a_m = \frac{\sum_{m' \in \mathcal{N}_m} w_{mm'}}{|\mathcal{N}_m|} \tag{1}$$

where $\mathcal{N}_m$ denotes the AAG neighbor set of annotator A_m. LAS values are normalized to $a'_m \in [0, 1]$, where values closer to 1 or 0 represent high or low annotator agreement, respectively. Normalized LAS values a'_m can be used to rank annotator A_m trustworthiness in descending order. They can also be used to perform a weighted voting data label aggregation. For each data entry $\mathbf{x}_n$ and for each label c_l, we calculate the *Weighted Aggregated Score (WAS)* L_{nl} of all annotators $A_m \in \mathcal{S}_n$:

$$L_{nl} = \sum_{y_{nm} = c_l} a'_m. \tag{2}$$

A unique label $c_{l'} \in \mathcal{C}$ is then assigned for each data entry $\mathbf{x}_n$ if there exists an $L_{nl'}$ such that:

$$L_{nl'} > \frac{L_{nl}}{2}, \; \forall l \neq l'. \tag{3}$$

Therefore, the most trustworthy class label $c_{l'}$ is assigned to data entry $\mathbf{x}_n$ if the total WAS score for that label exceeds half of the sum of the normalized LAS a'_m values for every label assigned by all $A_m \in \mathcal{S}_n$ annotators, other than $c_{l'}$. If there is no such class label that fullfils criterion (3) the data entry $\mathbf{x}_n$ is discarded. This rule ensures that the assigned label has strong support from reliable annotators and results in a unique label per data sample.

III. TMV Experimental Performance Evaluation

The *NetworkX* [16] Python library was used to construct the AGG graph.

To demonstrate the efficiency of the TMV method, we worked on text sentiment recognition. As human text sentiment labeling can be quite subjective, multiple human annotators were used to tag text with sentiment labels for DNN training and testing. The GoEmotions text dataset [17] comprising of $N = 58,000$ Reddit comments was annotated with 27 sentiments and a neutral emotion class label, totalling $L = 28$ class labels. Each text sample has been labeled by three upto five out of $M = 82$ unique annotators. A number of DNN training and testing experiments have been performed on this dataset for text sentiment recognition.

A. Experiments without label noise addition

GoEmotions labels, as a crowdsourced dataset, are intrinsically noisy, since text sentiment labeling is an inherently subjective task. The first experiment was to train a text sentiment recognition DNN using the proposed TMV label aggregation and compare its performance versus the one obtained using majority voting for training dataset labeling. Figure 1 depicts the AAG graph of 82 annotators produced by applying our TMV method on the GoEmotions dataset using threshold $T = 92$. While some annotators agree with each other (high LAS), others deviate from the majority (low LAS), essentially being outliers. For sentiment recognition, a RoBERTa [18] model was fine-tuned using the aggregated labels produced by the proposed TMV method and was compared against a baseline RoBERTa model that was fine-tuned using labels produced by simple majority voting. The text samples from the GoEmotions dataset were preprocessed following the approach of [19]. All experiments utilized the Transformers library [20].

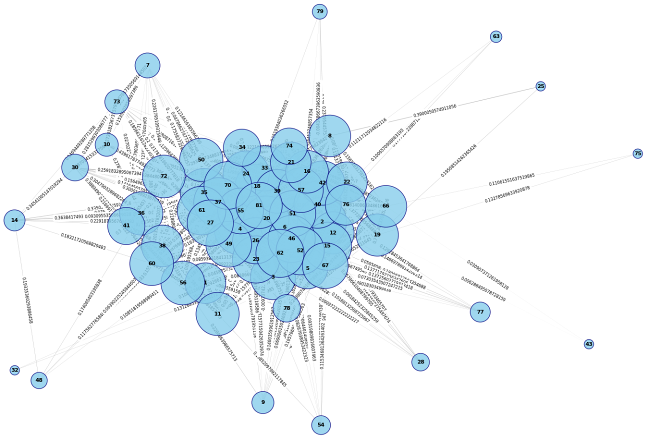

Fig. 1. AAG visualization for the 82 annotators that labeled the GoEmotions dataset.

To ensure a fair comparison with previous work, we initially refrained from applying our method to the test and validation splits of the GoEmotions dataset. In this configuration, our weighted label aggregation method was applied exclusively on the training dataset, resulting in a new training ground truth that was then used to train the RoBERTa sentiment classifier. Additionally, we also explored a second configuration where the proposed TMV method was applied to both the training and test datasets, allowing us to evaluate the model performance under conditions where both datasets were refined through the label aggregation process.

The DNN performance results on the test set, for both the aforementioned configurations are listed in Table I. F1-macro-weighted results are presented, as classification precision, recall, and accuracy can be misleading in multi-label classification tasks, especially in imbalanced datasets, such as GoEmotions. The results clearly show that RoBERTa classifier trained on our TMV aggregated labels greatly outperform those trained using simple majority voting across all text sentiment classes. Overall, the proposed method shows an approximate increase of 16.7% in the average F1-macro scores across all 28 classes, compared to the majority voting method for the GoEmotions dataset. It must be noted that the use of TMV on both the training and test data label aggregation is inferior to its use only on aggregating the training data set labels.

TABLE I
F1-MACRO COMPARISON BETWEEN TMV APPLIED EXCLUSIVELY ON A) THE TRAINING DATASET (TMV ON TRAIN LABELS), B) TMV APPLIED ON BOTH TRAINING AND TEST DATASETS (TMV ON TRAIN/TEST LABEL), C) AND MAJORITY VOTING (SMV).

Label	**TMV on Training Labels (a)**	**TMV on Training/Test Labels (b)**	**SMV on Training/Test Labels (c)**	**Difference (a)-(c)**
Admiration	**80.34%**	73.97%	70.80%	+9.54%
Amusement	**86.48%**	72.90%	83.20%	+3.28%
Anger	**73.86%**	69.61%	51.70%	+22.16%
Annoyance	**60.26%**	59.86%	34.90%	+25.36%
Approval	**62.76%**	60.05%	43.70%	+19.06%
Caring	**59.40%**	56.73%	40.50%	+18.90%
Confusion	**67.29%**	63.95%	47.00%	+20.29%
Curiosity	**68.19%**	65.04%	56.80%	+11.39%
Desire	**64.55%**	69.32%	52.40%	+12.15%
Disappointment	**64.23%**	60.78%	39.00%	+25.23%
Disapproval	**60.64%**	58.21%	43.90%	+16.74%
Disgust	**71.02%**	68.47%	49.10%	+21.92%
Embarrassment	**71.84%**	63.84%	50.70%	+21.14%
Excitement	**69.00%**	60.46%	45.50%	+23.50%
Fear	**84.30%**	69.03%	68.90%	+15.40%
Gratitude	**95.83%**	88.35%	92.20%	+3.63%
Grief	**49.98%**	49.95%	33.30%	+16.68%
Joy	**78.06%**	70.05%	63.40%	+14.66%
Love	**89.43%**	81.30%	81.20%	+8.23%
Nervousness	59.25%	**60.24%**	43.20%	+16.05%
Optimism	**77.01%**	70.93%	57.20%	+19.81%
Pride	**85.27%**	60.11%	58.30%	+26.97%
Realization	**60.69%**	55.15%	26.60%	+34.09%
Relief	49.96%	**58.44%**	24.60%	+25.36%
Remorse	**75.51%**	73.69%	68.80%	+6.71%
Sadness	**73.00%**	67.85%	59.10%	+13.90%
Surprise	**73.05%**	69.69%	60.10%	+12.95%
Neutral	**72.97%**	66.80%	68.80%	+4.17%
Average	70.86%	66.07%	54.10%	16.76%

Furthermore, it is evident that our method yields substantial improvements in F1-macro scores across a wide range of sentiment classes. For instance, some sentiment classes such as Annoyance, Disappointment, Realization, and Relief exhibit F1-macro increases that are greater than 25%. This manifests that our method is particularly effective in enhancing the recognition of more challenging or ambiguous sentiments. On the other hand, certain classes such as Neutral, Gratitude, and Amusement did not show a significant text sentiment analysis performance increase. This fact confirms that these text sentiment classes are inherently easier to classify. Hence simple majority voting is already relatively effective for creating annotator consensus on these labels.

Table II lists the balanced accuracy of loss-modeling method [11] and our proposed TMV method, when applied directly to the class labels of the GoEmotion dataset. We observe that our TMV approach surpasses previous state-of-the-Art (sotA) in all six Ekman sentiments [21] with an average increase of 7%.

B. Excessive Label Corruption Experiment

The label corruption experiment aims to evaluate the robustness and credibility of the proposed TMV method. We inject label corruption to a p_N ratio of the original unaggregated data labels, where $p_N \in [0, 1]$, resulting in $Kc = p_N N$

TABLE II
BALANCED SENTIMENT CLASSIFICATION ACCURACY COMPARISON OF TMV COMPARED TO LOSS-MODELING METHOD

Sentiment Class	**TMV**	**Loss-Modeling**	**% Increase**
Anger	**70%**	67%	4.4%
Disgust	**68%**	65%	4.6%
Fear	**73%**	70%	4.2%
Joy	**76%**	67%	13.4%
Sadness	**72%**	68%	5.8%
Surprise	**76%**	69%	10.1%

randomly selected text samples. For each corrupted sample x_k, $k = 1, \dots, K_c$ we randomly select a fraction $f \in [0, 1]$ of all y_{km} labels to be altered.

To simulate excessive label noise, we selected a value of p_N equal to 0.5 of the total GoEmotions samples and allocated new labels to each class for $f = 0.5$. Since the GoEmotions dataset comprises of 28 classes, this results in a heavily corrupted label set for each text sample containing up to 14 wrong sentiment labels ($f = 0.5$) This simulated noise injection is applied exclusively on the training GoEmotions dataset. Next, we aggregated the labels with TMV and SMV methods. We trained two separate RoBERTa [18] classifiers trained on TMV and SMV aggregated labels to evaluate the effectiveness of the proposed method under conditions of such an excessive label noise. Table III illustrates the average text sentiment recognition performance metrics for both scenarios.

TABLE III
COMPARISON OF TEST SET RESULTS BETWEEN 50% CORRUPTED LABELS (SMV) AND CORRECTED CORRUPTED LABELS WITH TMV .

Metric	SMV**(a)**	TMV**(b)**	**(b)-(a)**
Average Precision	3.16%	**3.89%**	+0.73%
Average Recall	-	**11.83%**	+11.83%
Average F1-macro	2.89%	**48.61%**	+45.72%
Average F1-micro	3.16%	**88.87%**	+85.71%
Average Accuracy	3.16%	**88.87%**	+85.71%

We observe that the RoBERTa model using SMV training label aggregation collapses under intense data corruption, as all sentiment classification metrics are at 3% level indicating extremely low SMV+DNN model performance. The proposed TMV method offers a great improvement compared to SMV. For example, its F1-macro performance metric exeeds the one of SMV 45.7%. Additionally, text sentiment recall and accuracy increased by 11.83% and 85.71% respectively, indicating that using the proposed TMV method demonstrates robustness under severe label noise.

C. Malicious Annotator Detection

To evaluate the efficiency of our method in identifying malicious or noisy annotators, we conducted two experiments: a) one to evaluate whether our proposed method identifies artificially injected malicious annotations and b) another one to evaluate the corruption intensity required to deem a trustworthy annotator as malicious or noisy one.

1) Artificial Malicious Annotator Detection: Our first experiment is designed to evaluate the effectiveness of our method in identifying malicious or noisy annotators. The experiment consists of the following two steps:

Inject an artificial new noisy annotator in the labeled dataset. As we already have 82 annotators (labeled 1-82), the new annotator ID is set to 83. This annotator allocates random labels to 10%, 15%, and 25% of the entire GoEmotions dataset. This simulates the behavior of an annotator who does not follow any consistent labeling pattern or exhibits malicious intent.

Find the annotator rankings and the new weighted aggregated labels. In all three corruption levels, our TMV method detected this annotator as noisy or malicious and assigned its normalized LAS score a'_m to zero. Figure 2 depicts the normalized LAS values a'_m of each annotator (having ID label 1-83) that satisfy the threshold of $T = 92$ commonly annotated texts as the annotator with ID 83 having a a'_m value equal to 0.

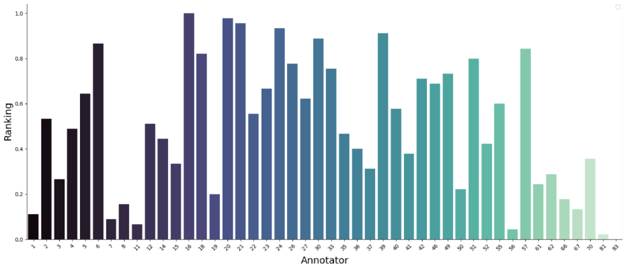

Fig. 2. Normalized LAS values, $a'_m, m = 1, \cdots, 82$ including a simulated malicious annotator at 15% label corruption level.

2) Existing Annotator Corruption: We selected an annotator with near-perfect rating (Annotator ID = 20, illustrated in Figure 2), whose normalized LAS value a'_m, $m = 20$ is close to 1. Then we randomly corrupted some of its labels, gradually increasing the label corruption frequency from 5% up to 90% with increments of 5%. The purpose of this experiment is to determine the level of label corruption that destroys the annotator trustworthiness, by significantly lowering its normalized LAS value a'_m.

Figure 3 shows how the annotator normalized LAS value a'_m decreases, as the label corruption percentage increases. We note its LAS value becomes zero at a 35% label corruption percentage. At this breakpoint, our TMV aggregation method disregards this annotator as a completely noisy/unstrusted one. It is important to note that the annotator normalized LAS value a'_m declines below the 0.1 level at a 25% label corruption frequency , much earlier than the aforementioned breakpoint of 35% corruption, resulting in a minimal contribution of this annotator during TMV label aggregation. On the positive side, label corruption levels up to 5% do not significantly impact the normalized LAS values a'_m and, hence, annotator trustworthiness.

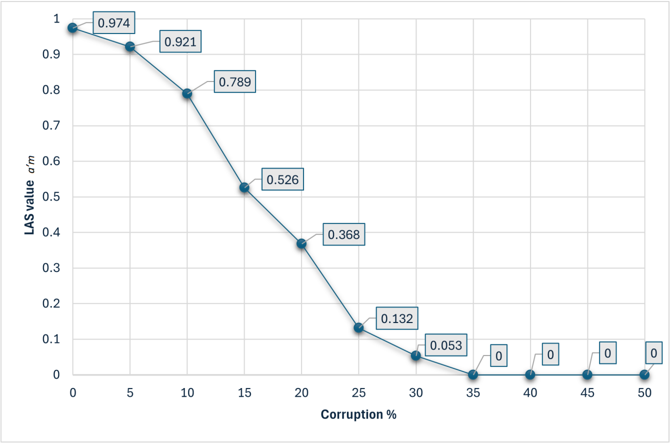

Fig. 3. Annotator normalized LAS value decrease versus label corruption frequency (%).

IV. Conclusions

In this work, we developed a method for label aggregation in multi-annotator datasets, specifically applied to the intrinsically subjective task of text sentiment classification. We demonstrated that our novel TMV aggregation method based on annotator trustworthiness outperforms SMV and Loss-Modeling ones, when used in text label aggregation for DNN-based text sentiment classification, it was proven that it effectively identifies added noisy or malicious annotators. To test the robustness of our method, we introduced varying levels of label corruption to an existing trustworthy annotator, creating a mix of high and low-quality text sentiment annotations. The proposed method successfully decreased the trustworthiness of the corrupted annotator, thereby reducing its final contribution in the labeling aggregation process of the training text data.

Furthermore, we evaluated our approach under conditions of excessive label-level corruption, altering up to 50% of the total annotations. RoBERTa models that were fine-tuned using our aggregation technique demonstrated superior performance across all evaluation metrics.

Looking forward, our proposed TMV method can be expanded for applications beyond text sentiment classification, to scenarios where annotations are subjective or prone to inconsistency, for instance in medical image segmentation or audiovisual sentiment detection.

Acknowledgement

This work has received funding from the European Commission—European Union under HORIZON EUROPE (HORIZON Research and Innovation Actions) under grant agreement No 101093003 (TEMA HORIZON-CL4-2022-DATA01-01). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union—European Commission. Neither the European Commission nor the European Union can be held responsible for them.

We thank P. Giannouris for his helpful comments while conducting this related on text sentiment analysis research.

References

[1] R. Snow, B. O'Connor, D. Jurafsky, and A. Ng, "Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks," in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, M. Lapata and H. T. Ng, Eds. Honolulu, Hawaii: Association for Computational Linguistics, 10 2008, pp. 254–263. [Online]. Available: https://aclanthology.org/D08-1027

[2] J. R. Jim, M. A. R. Talukder, P. Malakar, M. M. Kabir, K. Nur, and M. Mridha, "Recent advancements and challenges of nlp-based sentiment analysis: A state-of-the-art review," *Natural Language Processing Journal*, vol. 6, p. 100059, 3 2024.

[3] A. A. Chakrabarty, "Text data labelling using transformer based sentence embeddings and text similarity for text classification," *International Journal on Natural Language Computing*, vol. 11, pp. 1–8, 4 2022.

[4] X. Lu, "Learning ambiguity from crowd sequential annotations," *ArXiv*, vol. abs/2301.01579, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:255416226

[5] B. Parhami, "Voting algorithms," *IEEE Transactions on Reliability*, vol. 43, no. 4, pp. 617–629, 1994.

[6] N. Littlestone and M. Warmuth, "The weighted majority algorithm," *Information and Computation*, vol. 108, pp. 212–261, 2 1994.

[7] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," pp. 20–28, 1979.

[8] D. Cai, D. T. Nguyen, S. H. Lim, and L. Wynter, "Variational bayesian inference for crowdsourcing predictions," 6 2020. [Online]. Available: http://arxiv.org/abs/2006.00778

[9] D. Hovy, T. Berg-Kirkpatrick, A. Vaswani, and E. Hovy, "Learning whom to trust with mace," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 1120–1130.

[10] F. Rodrigues and F. Pereira, "Deep learning from crowds," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 04 2018.

[11] U. Jinadu, J. Annan, S. Wen, and Y. Ding, "Loss modeling for multi-annotator datasets," 11 2023. [Online]. Available: http://arxiv.org/abs/2311.00619

[12] A. M. Davani, M. Díaz, and V. Prabhakaran, "Dealing with disagreements: Looking beyond the majority vote in subjective annotations." [Online]. Available: http://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00449/1986597/tacl_a_00449.pdf

[13] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Unsupervised label noise modeling and loss correction," *ArXiv*, 4 2019. [Online]. Available: http://arxiv.org/abs/1904.11238

[14] S. Ibrahim, P. A. Traganitis, X. Fu, and G. B. Giannakis, "Learning from crowdsourced noisy labels: A signal processing perspective," 7 2024. [Online]. Available: http://arxiv.org/abs/2407.06902

[15] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, pp. 37–46, 4 1960.

[16] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conference*, Varoquaux, Gaël and Vaught, Travis and Millman, Jarrod, Ed., Pasadena, CA USA, 2008, pp. 11 – 15. [Online]. Available: http://conference.scipy.org/proceedings/SciPy2008/paper_2/

[17] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, and S. Ravi, "Goemotions: A dataset of fine-grained emotions," 5 2020. [Online]. Available: http://arxiv.org/abs/2005.00547

[18] Y. Liu *et al.*, "Roberta: A robustly optimized bert pretraining approach," 7 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[19] G. P. A. Mary, M. S. Hema, R. Maheshprabhu, and M. N. Guptha, "Sentimental analysis of twitter data using machine learning algorithms." Institute of Electrical and Electronics Engineers Inc., 2021.

[20] T. Wolf *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, 10 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[21] P. Ekman, "Are there basic emotions?" *Psychological Review*, vol. 99, pp. 550–553, 1992. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/1344638/