Audio-based Toxic Language Classification using Self-attentive Convolutional Neural Network

Conference Paper · August 2021

DOI: 10.23919/EUSIPCO54536.2021.9616001

2 authors:

Midia Yousefi

University of Texas at Dallas

16 PUBLICATIONS **194** CITATIONS

SEE PROFILE

Dimitra Emmanouilidou

Microsoft

48 PUBLICATIONS **577** CITATIONS

SEE PROFILE

Audio-based Toxic Language Classification using Self-attentive Convolutional Neural Network

Midia Yousefi*
department of Electrical Engineering
University of Texas at Dallas
Richardson, USA
midia.yousefi@utdallas.edu

Dimitra Emmanouilidou
Audio and Acoustics group
Microsoft Research
Redmond, USA
Dimitra.Emmanouilidou@microsoft.com

***Abstract*—The monumental increase in online social interaction activities such as social networking or online gaming is often riddled by hostile or aggressive behavior that can lead to unsolicited manifestations of cyberbullying or harassment. In this work, we develop an audio-based toxic language classifier using self-attentive Convolutional Neural Networks (CNNs). As definitions of hostility or toxicity can vary depending on the platform or application, in this work we take a more general approach for identifying toxic utterances, one that does not depend on individual lexicon terms, but rather considers the entire acoustical context of the short verse or utterance. In the proposed architecture, the self-attention mechanism captures the temporal dependency of the verbal content by summarizing all the relevant information from different regions of the utterance. The proposed audio-based self-attentive CNN model is evaluated on a public and an internal dataset and achieves 75% accuracy, 79% precision, and 80% recall in identifying toxic speech recordings.**

***Index Terms*—toxic language detection, self-attention, hate speech, sentiment detection, cyberbullying**

I. Introduction

Multiplayer online gaming is a world-wide growing social networking platform that provides entertainment, enjoyment, and engagement for its users [1]. However, since most of the online games are highly interactive and competitive, they have the potential to cause destructive interactions among gamers [2]. Cyberbullying [3], cyber-harassment [4], abuse [5], hate speech [6], and toxic language [7] are examples of common negative online behavior on different social networking platforms. To identify such detrimental online behavior, many social networking platforms employ approaches such as manual moderating and crowdsourcing [8]. However, these approaches may be inefficient and not scalable [9]. Therefore, there has been an urge to develop methods to automatically detect toxic content [10], [11].

In the past decade, a variety of methods and techniques have been proposed to detect toxic language. Google and Jigsaw launched a project called *Prospective*, which employs Machine Learning techniques to rate toxicity of text comments [12]. Since lack of public datasets has always been a challenge for this application, authors in [13] collected 15M comments from public accounts on Instagram to forecast the presence and intensity of hostility using linguistic features. In another study, Martens et al. used chat-logs of Multiplayer Online Battle Arena (MOBA) games to develop a text-based toxic language detection for online gaming [14]. More recently, a number of studies have explored toxicity detection by multi-modal means and interactions [11], [15], [16]. These studies collected and annotated large corpora that contain text embedded in images from various social networking platforms. Multiple deep learning approaches were then used to fuse the visual and textual information for detecting hate speech.

So far, most of the developed toxicity identification methods work with either text or text embedded in images, and research on audio and video-based methods is very scarce [17], [18]. This is because toxicity usually happens in discussions and comment sections of most social platforms. Information from audio-based modalities can, in turn, be converted into text information using a robust Automatic Speech Recognition (ASR) or image captioning systems. However, in scenarios where the recorded audio contains different background noise, reverberation, overlapping speech, different languages, and diverse accents, the performance of ASR system drops significantly [19], and the derived text can therefore be deemed unreliable. Besides, there are many acoustic, tonal and emotional cues that could be lost in the recognition process, resulting in a degraded performance.

To address the aforementioned issues, this work proposes an audio-based toxic language classifier using self-attentive CNN. To the best of our knowledge, this is the first audio-based toxicity classification system in the literature, which relies primarily on the acoustic modality to classify toxicity in speech. The contributions of our work are threefold, *(i)* an audio-based toxic language classifier is proposed, *(ii)* the effect of two different attention mechanisms is studied on the classification performance, *(iii)* the proposed architecture is evaluated on an internal toxic-based corpus and on the public dataset IEMOCAP, originally annotated for sentiment detection, showcasing generalization of the proposed architecture.

The reminder of this paper is as follows. In Section II details of the internal dataset are presented. The proposed architecture is introduced in Section III; experiments and results are explained in Section IV, followed by the conclusions of this work in Section V.

* The first author performed this work during an internship at Microsoft.

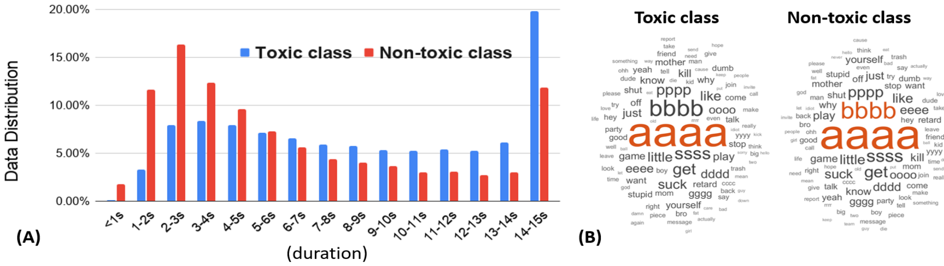

Fig. 1. (A) Utterance length. More than 20% of the utterances are < 4 sec long, mostly attributed to accidental recordings. (B) Word clouds.

II. Problem setup

The objective of this work is to identify whether a short audio-clip recording is toxic or not. For this purpose, toxicity has been defined as any language or tone that might discomfort the audience by containing traces of hate speech, direct bullying, or using a directly offensive language. Data used in this work come from online multi-player gaming platforms, which we call herein, *Corpus A*. Data comprised of short audio clips recorded during game playing, where users had the ability to report a part of the conversation as toxic behavior.

Each recording was then manually reviewed and labelled as either toxic or non-toxic by a human annotator. No refined annotations were available besides a single label per utterance. Out of all available audio clips 113,252 utterances were labelled as Toxic, and 25,660 as Non-toxic by an expert moderator. The duration of each recorded utterance could be arbitrary, up to a maximum of 15 sec (see Figure 1 (A)).

What is worth noting here, is the similarity in the word content among the two classes, an observation that further strengthens and motivates the proposed audio-based approach. Figure 1 (B) presents a top 100-word cloud visualization for this corpus. Utterances that were noisy or distorted, of foreign language or achieving low transcribing confidence were temporarily excluded for the creation of the word cloud. Text from transcribed speech was normalized for abbreviations, lemmatization, stop-, short- and long-word removal. Profanity here has been camouflaged in the form of letter sequences; for e.g. “aaaa” or “bbbb” refer to unique offensive words, and they refer to the same word for the two classes. One can see that identifying toxicity goes far beyond identifying individual swear words; contextual or situational information and other verbal cues are further needed for a better decision.

Finally, note that *Corpus A* comprises of naturalistic speech with utterances recorded by different users. Different microphones types, various room environments, background noises, background music, and overlapping speech introduce further challenges in the corpus, especially for any model based on ASR performance. An audio-based model seems essential for this type of work, whether part of a multi-modal solution or a standalone approach.

III. System design

The proposed method is depicted in Figure 2, in which toxicity classification is carried out in two steps: *(i)* extracting features mostly representative of toxic samples *(ii)* classifying them into toxic or non-toxic content. The first step is modeled using a CNN architecture that learns higher level information from the spectral features of speech. For the second step, we develop and configure Fully Connected (FC) layers to classify the extracted features. However, as previously noted, toxicity seems to manifest not just locally, but throughout a phrase/sentence, therefore a mechanism is needed to summarize the frame-level feature map into an utterance-level feature vector. The most straightforward approach to convert a feature map into a feature vector is to perform average pooling over time, depicted in Figure 2 as the baseline. Nonetheless, in many cases, not the entire content of an utterance is toxic. Hence, in scenarios where toxicity happens only in a small fraction of time, performing average pooling might decimate relevant temporal information. In such a scenario, despite the overall positive or neutral cues in an utterance, the content is still toxic, and an average pooling operator may wash out segments of interest. To tackle this challenge, an attention mechanism is integrated into the network to condense the feature map into a feature vector without losing relevant information. The effect of two alternate attention mechanisms calle “Learnable Query Attention” and “Self-Attention” are further studied on this task of toxicity identification.

Learnable Query Attention (LQ-Att) – the core idea of attention is to compress all the important information of a sequence into a fixed-length vector, so that computational resources can focus on a restricted set of important elements [20]. Attention finds the most informative regions in the feature map, then assigns reasonable weights to those regions [20]. To find relevant information and to calculate the dynamic weights for each time step, a (key, value) pair is defined as a linear transformation of the input [21]:

$$K = W_{key} \times X_{feat} \tag{1}$$

$$V = W_{value} \times X_{feat} \tag{2}$$

where, K and V stand for key and value respectively. W_{key} and W_{Value} are two learnable matrices that perform the linear transformation from input feature map X_{feat}. In addition to the (key, value) pair, attention needs an element known as Query to search for the relevant information in the input sequence. That is to say, Query is a pattern that we aim to find in the feature map, as a representation of toxicity. In this study, we define the Query as a trainable vector, so that the model learns a suitable representation throughout the optimization process. The attention output, depicted as the feature vector in Figure 2, is calculated as [21]:

$$Attention(q, K, V) = \text{softmax}(\frac{qK}{\sqrt{d_k}})V^T \tag{3}$$

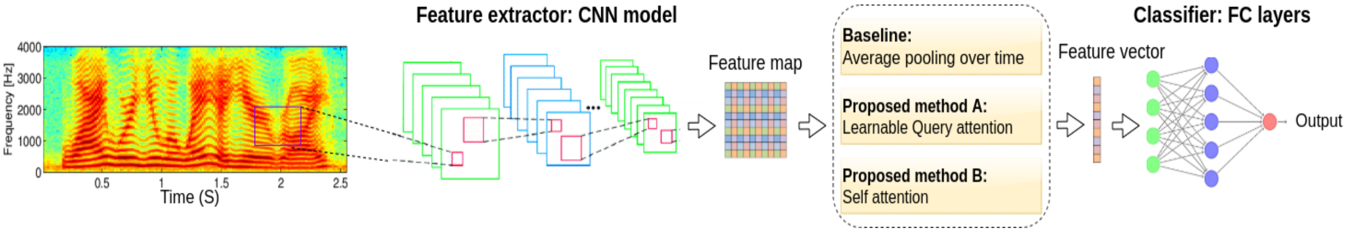

Fig. 2. The proposed architecture for audio-based toxic language classification.

where q is the trainable Query vector, and the scaling factor d_k is the dimension of the key K. According to equation 3, if a time step in the feature map has a key K similar to the Query q, the dot product of the corresponding key and Query will be high, which results in a larger weight for that specific time step. Next, the matrix value V is multiplied by the calculated attention weights, and then summed over the time dimension to form the output feature vector. Finally, the feature vector calculated at the output of attention is passed to the FC classifier followed by a Sigmoid activation function for the final decision. Although this approach is very practical, learning a robust Query might be very difficult. A weak Query may results in the loss of toxic-relevant information, which may impair the final decision.

Self-Attention (Self-Att) – this approach is proposed to tackle the challenge of learning a universally robust Query. Self-attention was first introduced in Neural Machine Translation [21], but it has also been very successful in abstractive summarization [22]–[24], and image description generation [25]. In Self-attention, different positions of a single sequence interact with each other to compute an abstract summary of the input sequence. Thus, in this way, Query is captured by the input sequence, by a linear transformation as:

$$Q = W_{query} \times X_{feat}, \tag{4}$$

In equation 4, Query Q is a matrix, meaning that each time step has an assigned Query vector q. For all possible combinations of two frames, say frame i and frame j, the query q_i of the first time step is compared to the key k_j of the second time step. The Softmax of the dot product of q_i and k_j is the attention weight α_{ij} which specifies how much the network should attend to region j while processing region i. Therefore, this approach is capable of capturing the entire context of the feature map and summarize it into a feature vector. Therefore, equation 3 is modified as:

$$Attention(Q, K, V) = softmax(\frac{QK}{\sqrt{d_k}})V^T \tag{5}$$

Self-attention is a powerful mechanism that generates the Query from the input ("self") and summarizes the entire information flow in the input sequence into a fixed-length feature vector.

IV. Experiments and Results

We first investigate the performance of the proposed methods on *Corpus A*, as introduced in Section II. To allow for low to moderate computational resources, we randomly select 20K utterances from both Toxic and Non-toxic classes while relatively restricting the utterance to within 4-8 seconds. The utterances were split into 15K for train (tr), 2.5K for cross-validation (cv), and 2.5K for testing (tt). Three separate sets of tr/cv/tt subsets were created by randomly shuffling the original utterances, creating 3 independent Monte Carlo runs. Average performance results are reported over all 3 runs.

Performance evaluation – the evaluation metrics used in this work are based on the confusion matrix: Accuracy (Acc), Weighted Accuracy (WAcc), Precision (Prec), Recall (Rec), and F-score (Fsc). Additionally, Receiver Operating Characteristic (ROC), Precision-Recall curve and the area under those two curves are reported for all the methods.

Model – Logarithmic Mel-Filter Banks (LMFB) were employed as the input of the model with audio data sampled at 16KHz. The 512-dim magnitude spectra were computed over a frame size of 25 ms with 10 ms of frame shift. A set of 40 triangular filters were introduced on the energy of the frame spectra and the logarithm of the output was calculated, comprising the final LMFB features.

We tuned the hyper-parameters of the baseline network using the cv subset. The choice of L = 4 2-D convolutional layers with C = 32 output channels, kernel size (K) of 5*5, and 2 FC layers with 256 neuron each is found optimum over a small parameter search of $L \in [3, 5]$, $K \in \{3, 5, 7\}$, and $C \in \{32, 64\}$. Kaiming initialization is used for all the layers in the experiments [26]. The output of the classifier is passed to a Sigmoid activation function for the final decision. The network parameters are updated by the the gradients of Binary Cross Entropy loss (BCEloss) using Stochastic Gradient Descent (SGD) optimizer with the initial learning rate LR = 0.01. The training process is completed by performing early stopping [27]. The maximum number of epochs is set to 200, batch size BS = 32 after a search in $BS \in \{32, 64, 128\}$; rate LR is set to a 0.7x decrease if the cv loss improvement is less than 0.001 for 2 successive epochs. No dropout layers were used. The early stopping is performed if no improvement is observed on the cv loss once the learning rate has decayed 4 times. The training and cross validation loss plots reveal a drastic drop during the first 10 epochs and commence plateau-ing after epoch 50 (not shown here), which depicts the ability of the network to generalize to unseen utterances in the development phase.

Table I shows the average performance on the 3 Monte Carlo runs. Performance of the Learnable-Query Att. is very close to the baseline, which may be expected due to the inexplicit or ambiguous nature of toxic content manifestation. The Self-Attention mechanism appears to learn more meaningful representations, noticeable by almost an 5% absolute

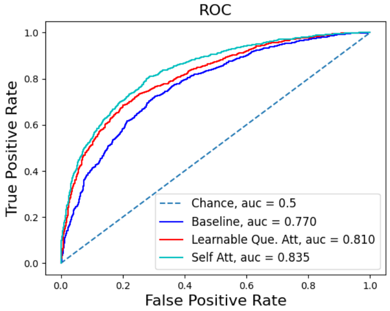

Fig. 3. ROC for Corpus A

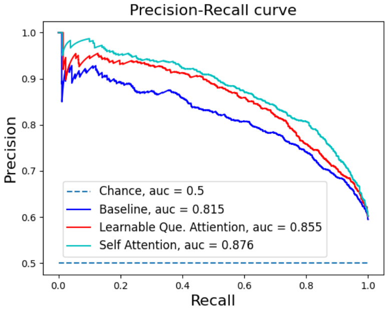

Fig. 4. Pre-Rec curve for Corpus A

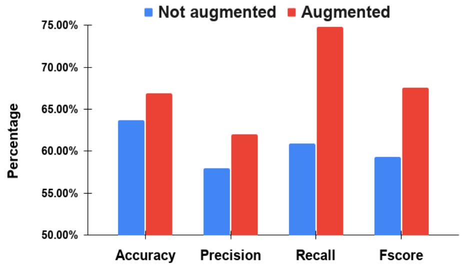

Fig. 5. Data augmentation on IEMOCAP

Corpus A	Acc	WAcc	Prec	Rec	Fsc
Baseline	71.33	69.36	73.87	79.96	76.79
LQ-Att	71.90	70.07	74.57	79.89	77.13
Self-Att	**75.87**	**74.80**	**79.16**	**80.51**	**79.82**

IEMOCAP	Acc	WAcc	Prec	Rec	Fsc
Baseline	66.87	66.58	62.05	74.83	67.58
LQ-Att	67.67	67.52	**68.07**	71.10	**69.54**
Self-Att	**68.85**	**68.79**	63.79	**73.74**	68.37
H$_{val-3}$				57.30	
P$_{val-3}$	64.45				
A$_{cat-4}$				71.80	

TABLE I
EVALUATION METRICS (%) FOR THE PROPOSED METHODS.

improvement on the weighted accuracy and precision. This improvement can be attributed to the ability of Self-Attention to summarize the entire content of the utterance into a single feature vector without missing on critical relevant information. The standard deviation for all systems and metrics ranges ∼0.8-2.2% (not shown). The ROC and Precision-Recall curves are depicted in Figures 3 and 4. The Area Under Curve (AUC) in both ROC and Precision-Recall curves for Self-Attentive CNN is relatively 7% higher than the baseline, which again, shows the ability of Self-Attention to capture relevant information. The input feature vectors and the feature vectors extracted using Self-Attentive CNN are visualized by PCA and t-SNE in Figure 6, where red and blue colors correspond to the two classes. The Self-Attentive CNN extracts higher-level features which clearly appear more divisible than the LMFB features in both PCA and t-SNE plots. The feature space seems more separable, attesting to a meaningful learnt representation for toxicity-related tasks.

The proposed work was also evaluated on the IEMOCAP corpus [28]. Although this dataset is on a different domain, i.e. sentiment analysis, we hope to provide i) a better demonstration on the effectiveness of the proposed architecture, and ii) a level of comparison by using a publicly available dataset. Available labels were adjusted to better resemble the previous setting: emotion categories of *happy* and *excited* were combined into a positive class, while *frustrated* and *angry* were merged into a negative class, using audio recordings from all scripted and improvised sessions. The recommended 5-fold cross validation was used for training and testing. Since IEMOCAP is a smaller dataset (3K utterances for training) compared to *Corpus A*, we augmented data with spectral Augmentation (SpecAug) [29]. The model hyperparameters were re-tuned based on the 5-fold validation. SpecAug improves the results of IEMOCAP by 3-14% across different performance metrics (Figure 5). The results of the proposed architectures on the augmented IEMOCAP are shown in Table I. Overall, both attention mechanisms outperform the baseline where LQ Att achieves higher performance compared to Self-Att. This could be due to the fact that (chosen) emotions may present less variability within a class when compared to a toxicity task, and a reliable fixed Query may be possible to learn. For completion, note that SpecAugm showed no significant benefits for *Corpus A*, arguably due to the rich acoustical variety of the realistic recordings.

A direct comparison with prior work on the combined two-class categorical problem was not possible to the best of our knowledge. Table I includes prior art on audio-based sentiment analysis on IEMOCAP. The reader is advised to interpret the cited work comparison with caution, since they address a slightly modified problem or number of classes. In [30] Han et. al demonstrate a VGG-based ordinal classifier achieving 57.30% in Unweighted Average Recall (UAR) for a 3-way valence rating on IEMOCAP (H$_{val-3}$). In [31] the authors report 64.45% unweighted accuracy on a 3-way valence classification, using an Adversarial Auto-Encoder framework. In [32], authors report 71.80% in UAR for a 4-way categorical classification (Happy, Sad, Angry, Neutral) using LMFB features and a deep NN architecture (A$_{cat-4}$).

V. Conclusion

In this work, we propose a Self-Attentive CNN architecture to detect toxic speech based on acoustical features. The Self-Attention mechanism compares the content of every possible pair of time steps in each utterance and calculates a weight according to the similarity of their content. Therefore, for processing each time step, the weighted information of other regions are taken into account. This approach helps with summarizing the entire feature map into a feature vector while preserving critical relevant information. We also show that learning a representation for toxicity can be challenging when using a trainable Query vector. This could be attributed to the variable, subjective, situational or unclear nature of what

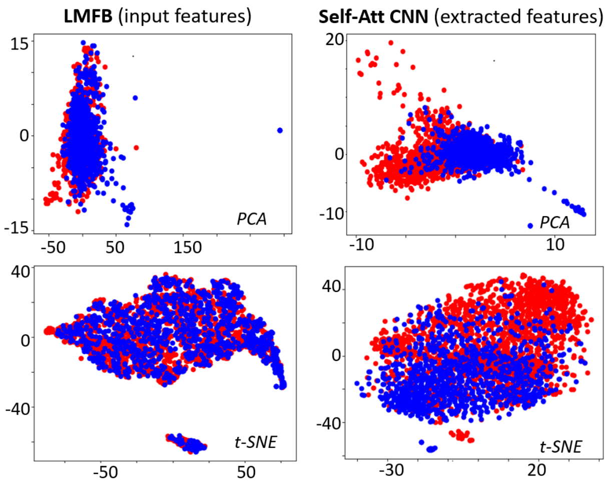

Fig. 6. Feature space separability visualization for Corpus A.

constitutes toxic content or behavior. Results showed that Self-Attention may boost the classification performance between Toxic and Non-Toxic utterances, by almost 5% absolute improvement for specific metrics, compared to the baseline. The AUC of the Precision-Recall curve also shows a relative improvement of 7%. The effectiveness of the proposed architecture is also studied on the public IEMOCAP corpus for the task of sentiment classification, which achieved a consistent absolute improvement of at least 2% over the baseline. Future work is needed to better understand the potential analogies and differences between audio-based toxicity and sentiment or affective domains, for further advancement of this field. The exploration of the supplemental value of text transcription also remains to be studied as future work.

References

[1] A. Tyack, P. Wyeth, and D. Johnson, "The appeal of moba games: What makes people start, stay, and stop," in *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, 2016, pp. 313–325.

[2] M. Griffiths, "Gaming in social networking sites: a growing concern?" *World Online Gambling Law Report*, vol. 9, no. 5, pp. 12–13, 2010.

[3] S. Salawu, Y. He, and J. Lumsden, "Approaches to automated detection of cyberbullying: A survey," *IEEE Transactions on Affective Computing*, 2017.

[4] T. Marwa, O. Salima, and M. Souham, "Deep learning for online harassment detection in tweets," in *2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS)*. IEEE, 2018, pp. 1–5.

[5] A. M. Founta, D. Chatzakou, N. Kourtellis, J. Blackburn, A. Vakali, and I. Leontiadis, "A unified deep learning architecture for abuse detection," in *Proceedings of the 10th ACM Conference on Web Science*, 2019, pp. 105–114.

[6] B. Gambäck and U. K. Sikdar, "Using convolutional neural networks to classify hate-speech," in *Proceedings of the first workshop on abusive language online*, 2017, pp. 85–90.

[7] A. Mayr, G. Klambauer, T. Unterthiner, and S. Hochreiter, "Deeptox: toxicity prediction using deep learning," *Frontiers in Environm. Science*, vol. 3, p. 80, 2016.

[8] J. Blackburn and H. Kwak, "Stfu noob! predicting crowdsourced decisions on toxic behavior in online games," in *Proceedings of the 23rd international conference on World wide web*, 2014, pp. 877–888.

[9] H. Chen, S. Mckeever, and S. J. Delany, "Harnessing the power of text mining for the detection of abusive content in social media," in *Advances in Computational Intelligence Systems*. Springer, 2017, pp. 187–205.

[10] V. K. Singh, S. Ghosh, and C. Jose, "Toward multimodal cyberbullying detection," in *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 2017, pp. 2090–2099.

[11] R. Gomez, J. Gibert, L. Gomez, and D. Karatzas, "Exploring hate speech detection in multimodal publications," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 1470–1478.

[12] "https://www.perspectiveapi.com//home."

[13] P. Liu, J. Guberman, L. Hemphill, and A. Culotta, "Forecasting the presence and intensity of hostility on instagram using linguistic and social features," *arXiv preprint arXiv:1804.06759*, 2018.

[14] M. Märtens, S. Shen, A. Iosup, and F. Kuipers, "Toxicity detection in multiplayer online games," in *2015 International Workshop on Network and Systems Support for Games (NetGames)*, 2015, pp. 1–6.

[15] T. Wijesiriwardene, H. Inan, U. Kursuncu, M. Gaur, V. L. Shalin, K. Thirunarayan, A. Sheth, and I. B. Arpinar, "Alone: A dataset for toxic behavior among adolescents on twitter," *arXiv preprint arXiv:2008.06465*, 2020.

[16] S. Suryawanshi, B. R. Chakravarthi, M. Arcan, and P. Buitelaar, "Multimodal meme dataset (multioff) for identifying offensive content in image and text," in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, 2020, pp. 32–41.

[17] A. Iskhakova, D. Wolf, and R. Meshcheryakov, "Automated destructive behavior state detection on the 1d cnn-based voice analysis," in *International Conference on Speech and Computer*. Springer, 2020, pp. 184–193.

[18] P. Alonso, R. Saini, and G. Kovács, "Hate speech detection using transformer ensembles on the hasoc dataset," in *International Conference on Speech and Computer*. Springer, 2020, pp. 13–21.

[19] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth'chime'speech separation and recognition challenge: dataset, task and baselines," *arXiv preprint arXiv:1803.10609*, 2018.

[20] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[22] K. Al-Sabahi, Z. Zuping, and M. Nadher, "A hierarchical structured self-attentive model for extractive document summarization," *IEEE Access*, vol. 6, pp. 24 205–12, 2018.

[23] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "Disan: Directional self-attention network for rnn/cnn-free language understanding," *preprint arXiv:1709.04696*, 2017.

[24] Y. Zhao, X. Ni, Y. Ding, and Q. Ke, "Paragraph-level neural question generation with maxout pointer and gated self-attention networks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3901–3910.

[25] P. Sharma, N. Ding, S. Goodman, and R. Soricut, "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 2556–2565.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE Inter. Conf. on Comp. Vision*, 2015, pp. 1026–34.

[27] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *preprint arXiv:1611.03530*, 2016.

[28] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, p. 335, 2008.

[29] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[30] W. Han, T. Jiang, Y. Li, B. Schuller, and H. Ruan, "Ordinal learning for emotion recognition in customer service calls," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 6494–98.

[31] S. Parthasarathy, V. Rozgic, M. Sun, and C. Wang, "Improving emotion classification through variational inference of latent variables," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 7410–14.

[32] Z. Aldeneh and E. M. Provost, "Using regional saliency for speech emotion recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 2741–45.