Toxic Language Identification Via Audio Using A Self-Attentive Convolutional Neural Networks (CNN)

P.Shyam Kumar [1], K.Anirudh Reddy [2], G.Kritveek Reddy [3], V. lingamaiah [4]
[1]Btech Student, Department of CSE, Anurag Group of Institutions, Hyderabad
[2]Btech Student, Department of CSE, Anurag Group of Institutions, Hyderabad
[3]Btech Student, Department of CSE, Anurag Group of Institutions, Hyderabad
[4]Assistant professor, Department of CSE, Anurag Group of Institutions, Hyderabad

Abstract:- The massive increase in online social interaction activities such as social networking and online gaming is frequently marred by hostile or aggressive behavior, which can result in uninvited manifestations of cyberbullying or harassment. In this paper, we use self-attentive Convolutional Neural Networks to build an audio-based toxic language classifier (CNNs). Because definitions of hostility or toxicity differ depending on the platform or application, we take a more general approach to identifying toxic utterances in this work, one that does not rely on individual lexicon terms, but rather takes into account the entire acoustical context of the short verse or utterance. The self-attention mechanism in the proposed architecture captures the temporal dependency of verbal content by summarizing all relevant information from different regions of the utterance. On a public and an internal dataset, the proposed audio-based self-attentive CNN model achieves 75% accuracy, 79% precision, and 80% recall in identifying toxicspeech recordings.

***Keywords*:-** *Toxic Language Detection, Self-Attention, Hate Speech, Sentiment Detection, Cyberbullying.*

I. INTRODUCTION

Online multiplayer gaming is a fast-growing social networking platform that offers users fun and excitement, gratification, and involvement. [1]. However, because most online games are highly interactive and competitive, they have the potential to cause harmful interactions between players [2]. Cyberbullying [3, 4], cyber-harassment [5, 6], abuse [7], hate speech [8, 9] are all examples of common negative online behaviour on various social networking platforms. Many social networking platforms use methods such as manual moderation and crowdsourcing to detect such harmful online behaviour [8]. These approaches, however, may be inefficient and not scalable [9]. As a result, there has been a push to create techniques for instantly detecting caustic substance [10], [11].

Several methods and techniques for detecting toxic language have been proposed over the last decade. Prospective is a joint Google and Jigsaw project that employs Machine Learning techniques to assess the toxicity of text comments. [12]. Because a lack of public datasets has always been a challenge for this application, the authors of [13] collected 15M comments from public Instagram accounts to forecast the presence and * The first author did this work while interning at Microsoft.

The intensity of hostility can be expressed using linguistic features. Martens et al. developed a text-based toxic language detection system for online gaming using chat logs from Multiplayer Online Battle Arena (MOBA) games in another study [14]. Recently, a number of studies have investigated multi-modal toxicity detection and interactions [11], [15], [16]. These studies gathered and annotated large corpora of text embedded in images from various social networking platforms. The visual and textual information was then fused using multiple deep learning approaches to detect hate speech.

So far, the majority of developed toxicity identification methods have relied on text or text embedded in images, with little research on audio and video-based methods [17], [18]. This is because most social platforms' discussions and comment sections are prone to toxicity. Audio-based modalities' information can then be converted into text information using a powerful Automatic Speech Recognition (ASR) system or image captioning systems. However, in situations where the recorded audio contains different background noise, reverberation, overlapping speech, different languages, and diverse accents, the ASR system's performance drops significantly [19], and the derived text can thus be deemed untrustworthy. Furthermore, many acoustic, tonal, and emotional cues may be lost during the recognition process, resulting in a degraded performance.

This paper proposes an audio-based toxic language classifier based on self-attentive CNN to address the aforementioned issues. This is, to the best of our knowledge, the first audio-based toxicity classification system in the literature that employs the acoustic modality to classify toxicity in speech.. Our work makes three contributions: I an audio-based toxic language classifier is proposed, (ii) the effect of two different attention mechanisms on classification performance is studied, and (iii) the proposed architecture is evaluated on an internal toxic-based corpus and on the public dataset IEMOCAP, which was originally annotated for sentiment detection, demonstrating generalization of the proposed architecture.

The following is a reminder of this paper. Section II presents specifics about the internal dataset. Section III presents the proposed framework; Section IV goes over the experiments and results; and Section V summarizes this work.

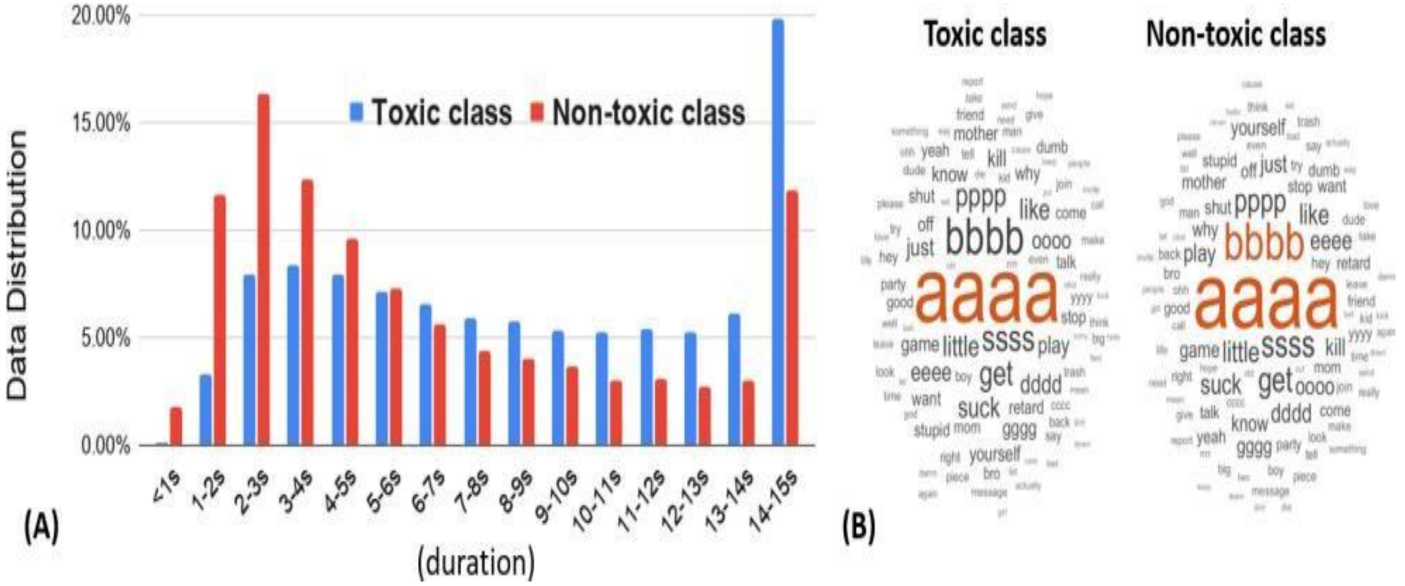

Fig 1 (A) Utterance length More than 20% of the utterances are < 4 sec long, mostly attributed to accidental recordings. (B) Word clouds. Information from spectral features of speech using a CNN

➢ *Setup*

The goal of this project is to determine whether or not a short audio clip recording is toxic. Toxic language or tone that contains traces of hate speech, direct bullying, or uses directly offensive language has been defined as toxicity for this purpose. The data used in this work comes from online multi- player gaming platforms, which we refer to as Corpus A. Dataconsisted of short audio clips recorded during game play, withusers able to report a portion of the conversation as toxic behavior.

A human annotator then reviewed each recording and labelled it as toxic or non-toxic. Apart from a single label per utterance, no refined annotations were available. An expert moderator labelled 113,252 utterances as Toxic and 25,660 as Non-toxic from all available audio clips. The length of each recorded utterance could be set arbitrarily, up to a maximum of 15 seconds (see Figure 1 (A).

What is important to note here is the similarity in word content between the two classes, an observation that reinforces and motivates the proposed audio-based approach. Figure 1 (B) shows a visualization of the top 100 words in thiscorpus. Utterances that were noisy or distorted, of foreign language, or had low transcribing confidence were temporarily excluded from the word cloud creation. Text from transcribed speech was normalized for abbreviations, lemmatization, and the removal of stop-, short-, and long- words. Profanity has been disguised as letter sequences; for example, "aaaa" or "bbbb" refer to distinct offensive words, and they refer to the same word for the two classes. As can be seen, identifying toxicity extends far beyond identifying specific swear words; contextual or situational information, as well as other verbal cues, are also required for a better decision.

Finally, keep in mind that Corpus A is made up of naturalistic speech with utterances recorded by various users. Different phone types, different room environments, background noises, background music, and overlapping speech all add to the corpus's challenges, especially for any model based on ASR performance. An audio-based model appears to be necessary for this type of work, whether as part of a multi-modal solution or as a standalone approach.

II. SYSTEM DESIGN

Figure 2 depicts the proposed method, which divides toxicity classification into two steps: *(i)* extracting higher-level features mostly representative of toxic samples *(ii)* classifying architecture. To classify the extracted features, we develop and configure Fully Connected (FC) layers in the second step. However, as previously stated, toxicity appears to manifest not only locally, but throughout a phrase/sentence, necessitating the development of a mechanism to summarize the frame-level feature map into an utterance-level feature vector. The most straightforward method for converting a feature map to a feature vector is to perform average pooling over time, which is depicted as the baseline in Figure2. However, in many cases, the entire content of an utterance is not toxic. As a result, in scenarios where toxicity occurs only for a short period of time, performing average pooling may decimate relevant temporal information. In such a case, regardless of whether an utterance contains overall positive or neutral cues, the content is still toxic, and an average pooling operator may wash out segments of interest. To address this issue, the network incorporates an attention mechanism that condenses the feature map into a feature vector while retaining relevant information. On this task of toxicity identification, the effect of two alternate attention mechanisms called "Learnable Query Attention" and "Self-Attention" is being investigated further.

➢ *Learnable Query Attention (LQ-Att)*

The basic ideabehind attention is to compress all of the important information in a sequence into a fixed-length vector, allowing computational resources to focus on a limited set of important elements [20]. Attention locates the most informative regions in the feature map and assigns appropriate weights to those regions [20]. A (key, value) pair is defined as a linear transformation of the input [21] to find relevant information and calculate dynamic weights for each time step:

$$K = W_{key} \times X_{feat} \qquad (1)$$

$$V = W_{value} \times X_{feat} \qquad (2)$$

Where, The letters K and V hold for key and value, respectively. *Wkey* and *WV alue* are two learnable matrices that perform the lineartransformation from input feature map *Xfeat*. 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 findin 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]:

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

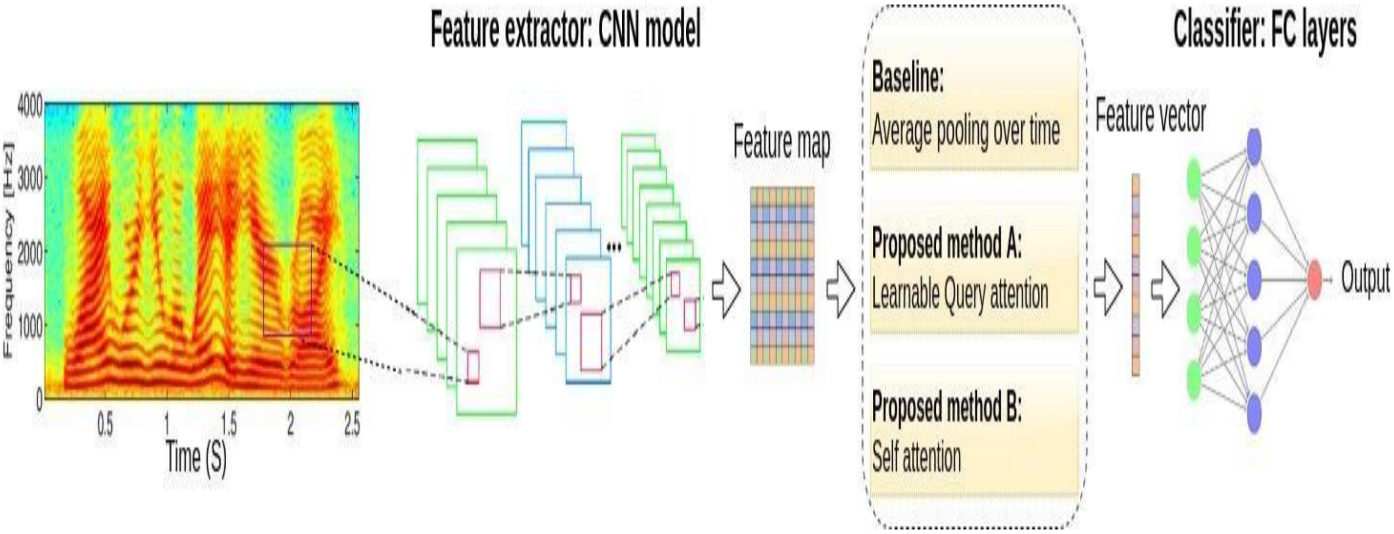

Fig 2 The Proposed Architecture for Audio-Based Toxic Language Classification

Where q denotes the trainable Query vector and *dk* denotes the dimension of the key K. If a time step in the feature map has akey K that is similar to the Query q, the dot product of the corresponding key and Query will be large, resulting in a larger weight for that time step. The output feature vector is then formed by multiplying the matrix value V by the calculated attention weights and summing over the time dimension. Finally, for the final decision, the feature vector calculated at the output of attention is passed to the FCclassifier, which is followed by a Sigmoid activation function. Although this method is very practical, mastering a robust Query may be difficult. A faulty query can lead to the loss of toxic-relevant information, which can skew the final decision.

➢ *Self-Attention (Self-Att)* – This method is proposed to address the problem of learning a universally robust Query. Self-attention was first introduced in Neural MachineTranslation [21], but it has also proven to be very effective in abstractive summarization [22]-[24] and image descriptiongeneration [25 In Self-attention, different places of a single sequence interact with one another to calculate a conceptual overview of the input sequence. Thus, Query is captured by the input pattern via a linear function as follows:

$$Q = W_{query} \times X_{feat}, \qquad (4)$$

In equation 4, Query Q is a matrix, which means that Queryvector q is assigned to each time step. The query *qi* of the first time step is compared to the key *kj of* the second time step for all possible combinations of two frames, say frame i and frame j. The attention weight ij is the Softmax of the dot product of qi and kj, which specifies how much the network should pay attention to region j while processing region i. As a result, this method can capture the entire context of the feature map and summarize it into a feature vector. As a result, equation 3 is changed to: keeping the utterance to a maximum of 4-8 seconds. The utterances were divided into three groups: 15K for training (tr), 2.5K for cross-validation (cv), and 2.5K for testing (tt). By randomly shuffling the original utterances, three distinct sets of tr/cv/tt subsets were generated, resulting in three independent Monte Carlo runs. The average performance results from all three runs are reported.

➢ *Performance evaluation*

The following evaluation metrics are based on the confusion matrix: Accuracy (Acc), Weighted Accuracy (WAcc), Precision (Prec), Recall (Rec), and F-score (Fsc).

Additionally, for each method, the Receiver Operating Characteristic (ROC), Precision-Recall curve, and area under those two curves are reported.

➢ *Model*

The model's input was Logarithmic Mel-Filter Banks (LMFB), with audio data sampled at 16KHz. The 512-dim magnitude spectra were calculated over a 25-ms frame size with a 10-ms frame shift. The energy of the frame spectra was passed through a set of 40 triangular filters, and the logarithm of the output, which included the final LMFB features, was calculated.

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 highest amount of epochs is set to 200, and the batch size BS is set to 32 after a search in BS 32, 64, 128; the rate LR is set to a 0.7x decrease if the cv loss improvement is less than 0.001 for two consecutive epochs.No dropout layers were used. The early stopping is performed if no improvement.

$$\text{Attention}\ (Q,\ K,\ V) = \text{softmax}\ \left(\frac{\sqrt{\_}V^{T}}{d_k}\right) \qquad (5)$$

Self-attention is a powerful mechanism for generating the Query from the input ("self") and encapsulating the entire information flow in the input sequence in a fixed-length feature vector.

III. EXPERIMENTS AND RESULTS

We begin by testing the performance of the proposed methods on Corpus A, as described in Section II. In order to accommodate low to moderate computational resources, We chose 20K utterances at random from both the Toxic and Non-toxic classes while 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.

The average performance on the three Monte Carlo runs is shown in Table I. Because toxic content manifestation is inexplicit or ambiguous, performance of the Learnable- Query Att. is very close to the baseline. The mechanism of Self-Attention appears to learn more meaningful representations.

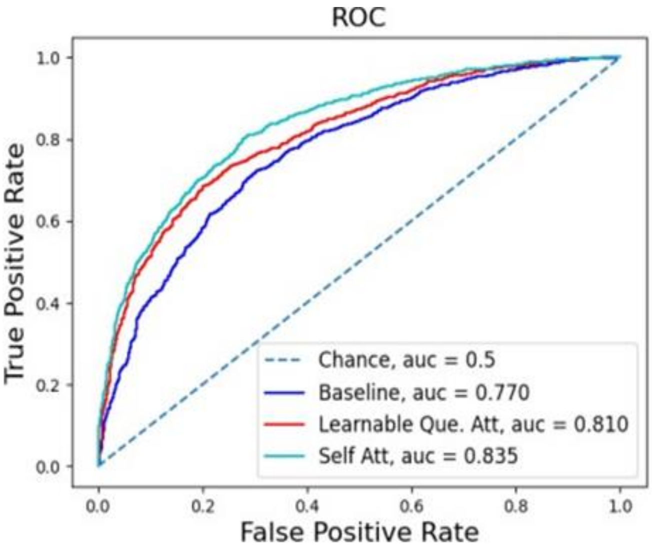

Fig 3 ROC for Corpus A

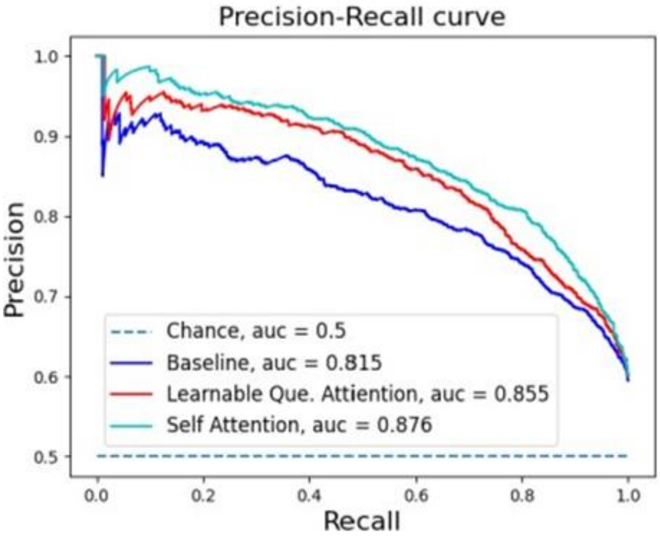

Fig 4 Pre-Rec Curve for Corpus A

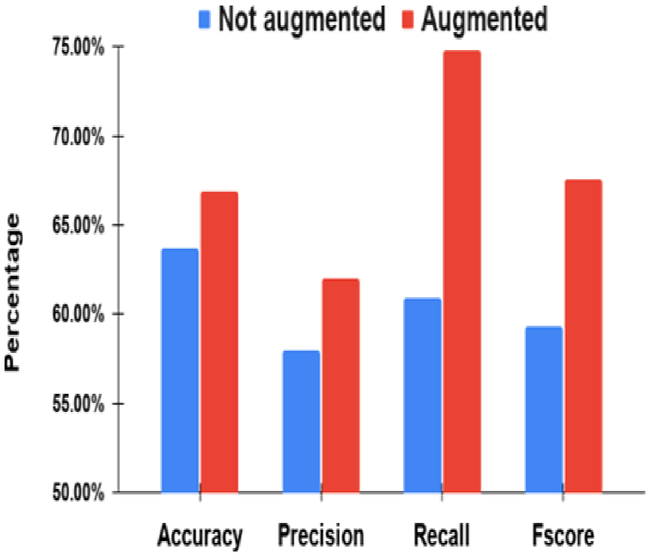

Fig 5 Data Augmentation on IEMOCAP

Table 1 Evaluation Metrics (%) For The Proposed Methods

Corpus A	**Acc**	**W Acc**	**Prec**	**Rec**	**Fsc**
Baseline	71.33	69.36	73.87	79.86	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**	**W Acc**	**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
Hval-3				57.30	
Pval-3	64.45				
Acat-4				71.80	

There is a nearly 5% absolute improvement in weighted accuracy and precision. This enhancement can be attributed to Self-ability Attention's to summarize the entire content of the utterance into a single feature vector while not missing out on critical relevant information. For all systems and metrics, the standard deviation ranges from 0.8 to 2.2%. (not shown). Figures 3 and 4 show the ROC and Precision-Recall curves. The Area Under Curve (AUC) for Self-Attentive CNN is 7% higher than the baseline in both the ROC and Precision-Recall curves, demonstrating the ability of Self-Attention to capture relevant information. Figure 6 depicts PCA and t-SNE visualizations of the input feature vectors and the feature vectors extracted using Self-Attentive CNN, where red and blue colors correspond to the two classes. In both PCA and t- SNE plots, the Self-Attentive CNN extracts higher-level features that are clearly more divisible than the LMFB features. The feature space appears to be more separable, indicating that a meaningful learnt representation for toxicity- related tasks exists.

On the IEMOCAP corpus, the proposed work was also evaluated [28]. Despite the fact that this dataset is in a different domain, sentiment analysis, we hope to provide I a better demonstration of the effectiveness of the proposed architecture and ii) a level of comparison by using a publicly available dataset. Using audio recordings from all scripted and improvised sessions, 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 combined into a negative class. For training and testing, the recommended 5-fold cross validation was used.

Because IEMOCAP is a smaller dataset (3K utterances for training) than Corpus A, we augmented it with spectral Augmentation (SpecAug) [29]. Based on the results of the 5- fold validation, the model hyperparameters were fine-tuned. SpecAug improves IEMOCAP results by 3-14% across multiple performance metrics (Figure 5). The results of the proposed architectures on the augmented IEMOCAP are shown in Table I. In general, both attention mechanisms outperform the baseline, with LQ Att outperforming Self-Att. This could be due to (selected) emotions having less variability within a class than a toxicity task, and a reliable fixed Query being learnable.

A proper comparison with previous work on the combination of two categorical problem was not possible, to the best of our knowledge. Prior art on audio-based sentiment analysis on IEMOCAP is included in Table I. Because they address a slightly modified problem or number of classes, the reader is advised to interpret the cited work comparison with caution. In[30] Han et. al show a VGG-based ordinal classifier that achieves 57.30% Unweighted Average Recall (UAR) for IEMOCAP gives a 3-way valence rating (Hval3). In [31] a 3-point scale, the authors report an unweighted accuracy of 64.45%. Using an Adversarial Auto-Encoder framework, we classified valence in three ways. The authors of [32] report a UAR of 71.80% for a 4-way. Acat4 uses LMFB features and a deep NN architecture to perform categorical classification (Happy, Sad, Angry, Neutral).

IV. CONCLUSION

We present a Self-Attentive CNN architecture for detecting toxic speech using acoustical features in this paper. The Self-Attention method contrasts the information of every possible pair of time steps in each pronouncement and assigns a weight based on their content similarity. As a result, for each time step, the weighted information from other regions is considered. This method aids in summarizing the entire feature map into a feature vector while preserving critical relevant information. We also show that when using a trainable Query vector, learning a representation for toxicity can be difficult.

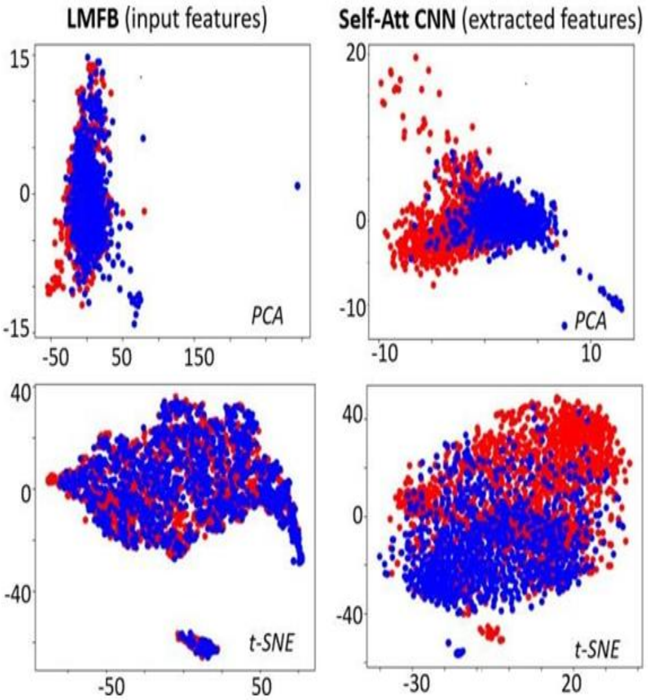

Fig 6 Feature Space Separability Visualization for Corpus A

This could be attributed to the variable, subjective, situational, or ambiguous nature of what constitutes toxic content or behaviour. The results showed that self-attention can improve classification performance between toxic and non-toxic utterances by nearly 5% absolute improvement for specific metrics when compared to the baseline. The AUC of the Precision-Recall curve has also improved by 7%. The proposed architecture's effectiveness is also tested on the public IEMOCAP corpus for sentiment analysis, which accomplished a consistent best of at least 2% over the baseline. To advance this field, more research is required to better understand the potential analogies and differences between voice toxicity and perception or affective outcomes. The examination of the extra value.

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 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. Gambä ck 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 deci- sions 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 Ab- stracts 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, "Mul- timodal 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 *Inter- national Conference on Speech and Computer*. Springer, 2020, pp. 184–193.

[18]. P. Alonso, R. Saini, and G. Kova´cs, "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: Direc- tional 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 cap- tioning," 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, “Un- derstanding 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 emo- tion 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 emo- tion recognition,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 2741–45.