Towards A Multi-agent System for Online Hate Speech Detection

Gaurav Sahu
University of Waterloo
Waterloo, Canada
gaurav.sahu@uwaterloo.ca

Robin Cohen
University of Waterloo
Waterloo, Canada
rcohen@uwaterloo.ca

Olga Vechtomova
University of Waterloo
Waterloo, Canada
ovechtom@uwaterloo.ca

ABSTRACT

This paper envisions a multi-agent system for detecting the presence of hate speech in online social media platforms such as Twitter and Facebook. We introduce a novel framework employing deep learning techniques to coordinate the channels of textual and image processing. Our experimental results aim to demonstrate the effectiveness of our methods for classifying online content, training the proposed neural network model to effectively detect hateful instances in the input. We conclude with a discussion of how our system may be of use to provide recommendations to users who are managing online social networks, showcasing the immense potential of intelligent multi-agent systems towards delivering social good[1].

KEYWORDS

Multi-agent System, Online Abuse Detection, Deep Learning, Multimodal Fusion

ACM Reference Format:
Gaurav Sahu, Robin Cohen, and Olga Vechtomova. 2021. Towards A Multi-agent System for Online Hate Speech Detection. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), London, UK, May 3–7, 2021*, IFAAMAS, 9 pages.

1 INTRODUCTION

A social problem that has become prominent of late is that of hate speech online. In this paper, we look at an approach for detecting instances of these kinds of posts, in contexts such as discussion boards and social networks. In particular, we examine how designing artificial intelligence algorithms which look at the interplay between visual content and textual content may be especially valuable. Each channel's processing algorithm can be viewed as an intelligent agent, and determining how best to coordinate the output of these channels into one cohesive interpretation is the multiagent challenge.

In this paper, we sketch an approach to this problem grounded in the techniques of deep learning. We contrast this vision with that adopted by other researchers examining how to automate the procedure of hate speech detection. We also discuss experiments which can demonstrate the advantages of our approach, considering the task at hand as one of effectively classifying the input (either hate speech or not).

From here we move on to reflect on how a multiagent viewpoint of the task of detecting online hate speech also lends itself to processing solutions which can be attuned to the user at hand. In essence, the intelligent agents can incorporate into their decision making some factors which capture the user's preferences and needs.

Connecting with the theme of this workshop, our research aims to address the uncertainties that arise within social networking environments, where new content is posted in real time and needs to be assessed and filtered if necessary in a timely manner. The benefit to society that we strive to achieve is to enable platform owners to remove harmful content, thereby protecting individuals and preventing online groups from propagating hate speech.

We introduce relevant background for this work in Section 2, formally describe the problem and proposed approach in Section 3, and showcase experiments in Section 4. Finally, Section 5 discusses extended applications of our framework and includes our concluding remarks and scope for future developments in this line of research.

2 BACKGROUND

Our proposed multi-agent system uses various artificial intelligence techniques and algorithms to extract features from multimodal input and detect hate speech. In the rest of this section, we discuss relevant background for this work.

2.1 Deep Learning

Deep Learning (DL) is a class of machine learning (ML) algorithms that employs multiple layers to extract features from raw input. Traditional ML algorithms rely on single-layered, simpler architectures such as support vector machines (SVMs) [6] and the probabilistic Naive Bayes model to extract features from given input; hence, they are unable to capture deep semantic and syntactic information residing in the input. The last decade has seen a tremendous growth in DL and its applications, and numerous classes of deep neural networks have been proposed to extract features from different types of inputs. We discuss prominent classes of neural networks proposed to process raw textual and visual input.

Recurrent Neural Networks (RNNs). A vanilla RNN is a network of *neurons* arranged into successive layers. Its nodes are connected to form a directed graph along a temporal sequence. The connections are uni-directional, and every neuron has a real-valued activation function. Their internal state, better known as *memory*, allows them to process inputs of variable length. A sentence can be expressed as a variable-length temporal sequence of words, where each word denotes one time-step; hence, RNNs have been extensively employed for various natural language processing tasks such as machine translation and text classification [9, 20, 47]. Long short-term memory (LSTM), a unique RNN that has feedback connections in addition to the standard memory cells is the most widely applied RNN for NLP tasks [8, 16].

[1]Corresponding author: Gaurav Sahu (gaurav.sahu@uwaterloo.ca)

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, London, UK.

Convolutional Neural Networks (CNNs). Convolutional neural networks (CNNs) are most popularly used for visual analysis [17, 24, 44, 48]. As the name suggests, CNNs use the convolution operation in at least one of their layers instead of general matrix multiplication in a vanilla multi-layered perceptron (MLP). While MLPs are prone to over-fitting due to their fully-connected nature, CNNs use small and simple patterns to learn bigger, more complex patterns in data. This results in fewer connections and lower complexity, making CNNs highly suitable for processing images and videos. AlexNet [25] and VGG [44] are two such popular CNNs that gave state-of-the-art performance on the ImageNet classification. Their deep hierarchical architecture allows them to capture an image's semantic information at various feature-levels.

Generative Adversarial Networks (GANs). GANs are a specialized class of deep generative models that learn to generate novel data samples from random noise, while matching a given data distribution [12]. For instance, a GAN trained on a dataset of anime character images can generate novel anime characters, which look highly authentic, at least superficially [4]. Their flexibility and a wide-range of applications make GANs a popular choice for generation. Moving past the originally proposed unsupervised learning regime, countless GAN variants successfully adapt it for semi-supervised learning [42], fully-supervised learning [18], and even reinforcement learning [15]. More advanced applications of GANs include modding in video games [52], motion analysis in video [51], and super-realistic image generation [22]. They have even been employed for many text generation tasks such as text style transfer [19, 55]. GANs also demonstrate impressive performance for multimodal tasks such as image captioning [34].

2.2 Multimodal Deep Learning

Multimodal Deep Learning (MMDL) involves relating features from multiple modalities (or modes) – the different sources of information – such as images, audio, and text. The goal is to learn a shared representation of the inputs from different modalities, which a neural network may exploit to make intelligent decisions for a desired task. The earliest attempts to develop such a system involve the work by Ngiam et al. [36], where sparse Restricted Boltzmann Machines (RBMs) and deep autoencoders were employed to demonstrate the benefit of introducing information from different sources. Their end-to-end deep graph neural network could reconstruct missing modalities at inference time. They also demonstrate that better features for one modality can be learned if relevant data from different modalities is available at training time. However, they also point out that their models could not fully capitalize on the existing information due to heterogeneity in multimodal data. A line of research dedicated to addressing this issue studies different *fusion* mechanisms–techniques to combine (or *fuse*) information from different modes. Earlier models such as the bimodal RBMs and Deep Boltzmann Machines (DBMs) use concatenation to fuse cues from different input modes. While it is a first step towards combining multimodal cues, it results in a shallow architecture [36, 46].

Zadeh et al. [58] and Liu et al. [32] use Cartesian product and low-rank matrix decomposition instead of concatenation. Tsai et al. [49] propose multimodal transformers (MulT) which use cross-modal attention mechanisms to combine multimodal cues. These mitigate the shallowness of the network, while capturing inter- and intra- modal dynamics simultaneously. However, the resultant architecture either poses a significant computational overhead or further adds to the complexity of a fusion model. We use GAN-Fusion and Auto-Fusion, two adaptive fusion mechanisms that outperform their massive counterparts on challenging multimodal tasks [41], described in more detail in Section 3.

2.3 Hate Speech

Social media platforms have enabled people to connect with others and readily share information. However, the malicious intent of a few individuals has created a toxic environment online. One of the prominent social media platforms, Twitter, defines hate speech as follows:

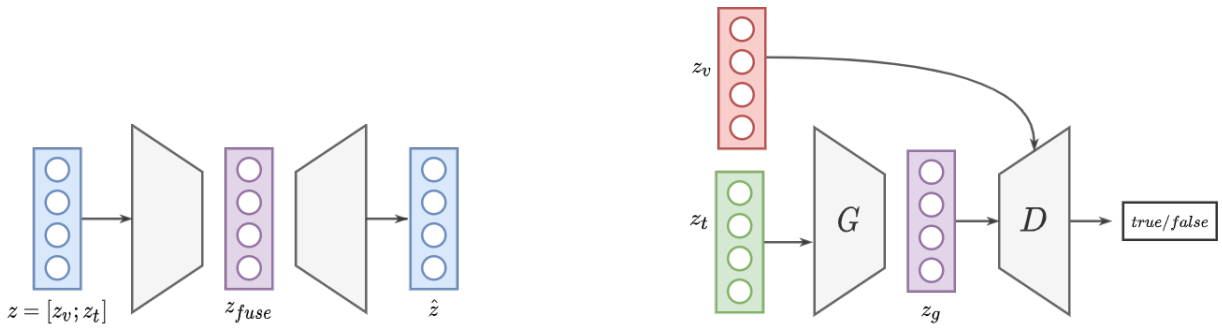
"Violence against or directly attack or threaten other people on the basis of race, ethnicity, national origin, sexual orientation, gender, gender identity, religious affiliation, age, disability, or serious disease."

It is, therefore, imperative for such platforms to have an autonomous agent that can flag potential instances of hate speech online. Such agents can aid in regulating the flow of such content on social media. Researchers in the AI community have focused on building solutions to address the issue.

Unimodal setting. Schmidt and Wiegand [43] enumerate automated hate speech detection systems in a unimodal setting, where the models leverage textual features such as bag-of-words, word embeddings, and dependency parsing information to detect presence of hate speech online [5]. Kwok et al. [27] and Greevy et al. [13] identify the domination of race and sex-based hate speech. Although surface-level features such as n-grams prove helpful, content on social media poses nuanced linguistic challenges like deliberate text obfuscation. For instance, "1d10t" is understandable for a human, but it can bypass algorithms relying on keyword spotting to detect hate speech in text [37]. Additionally, the web is full of different but effectively same words. For instance, "wow!" and "wooow!!" carry the same meaning. Such noisy tokens easily explode the vocabulary size and unnecessarily increase the task's complexity. Kumar et al. [26] investigate the use of neural attention mechanisms and gauge the effects of pre-processing to reduce out-of-vocabulary (OOV) instances.

Multimodal setting. Recently, there have been steady developments towards exploring hate speech detection in a multimodal context. Kiela et al. [23] released The Hateful Meme Challenge, and Gomez et al. [11] proposed the MMHS150K dataset. Both datasets are comprised of an image, its captioned text, and the task is to detect if the image exhibits hate speech or not. Zhu [59], Muennighoff [35], and Velioglu and Rose [50] explore the application of visual-linguistic transformers to extract meaningful cues from images and flag instances of hate speech. They also confirm the dominance of racist and sexist instances compared to other types of hate speech. However, these models employ late fusion techniques such as majority voting, which are known to ignore the inter-modal dynamics [58].

In the next section, we describe our approach, which addresses various issues related to hate speech and multimodal fusion.



(a) Auto-Fusion

(b) GAN-Fusion module for text

Figure 1: (a) Auto-Fusion module: First, z (concatenation of z_v and z_t) is passed through the autoencoder, which uses z_{fuse}, the intermediate latent vector, to output $\hat{z}$, a reconstruction of input z. (b) GAN-Fusion module for text: z_t is passed through a generator (along with a random normal noise), which tries to match its output z_g to z_v. The discriminator tries to guess the source of its input and outputs a *true/false* label.

3 METHODOLOGY

We refer to social media posts such as tweets and Facebook posts as *publications*. Therefore, a publication may consist of an image, text, or a combination of both.

Given an online publication p, we denote its visual and textual components by p_v and p_t, respectively. We pose multimodal hate speech detection as a classification problem: given a publication p, classify whether it exhibits hate speech or not. For simplicity, we focus on binary classification in this section, but depending on the experimental settings, the model can very easily be extended for a multi-class classification task.

The following subsections elaborate on different vital components of a multi-agent system to detect hate speech.

3.1 Visual (v) and Textual (t) Encoders

Given a raw multimodal publication p, we first encode p_v and p_t to learn meaningful vectorized representations. These encoders also serve as the first-level feature extractors.

We use a CNN to encode p_v. More specifically, we use a VGG [44] module, pre-trained on the ImageNet [7] dataset. Before processing p_v, we first transform its dimensionality such that it is a compatible input for the VGG module. Refer to Section 4 for more experimental details. We use an RNN to encode p_t. In particular, we use an LSTM cell because it helps in capturing long-range context in a sentence [16]. We also add a layer of word attention mechanism [33] to allow the model to pick up on more important words in a sentence. The visual and textual encoders output a fixed-dimensional latent vector, denoted by $\boldsymbol{z_v}$ and $\boldsymbol{z_t}$, respectively. Notably, $\boldsymbol{z_v}$ and $\boldsymbol{z_t}$ have an equal number of dimensions.

3.2 Entity Extraction Modules

As pointed out in Section 2, content on social media is noisy, which may deteriorate the quality of features extracted by the video and text encoders. Therefore, we use entity extraction modules to both remove noise from the input and learn a second set of features.

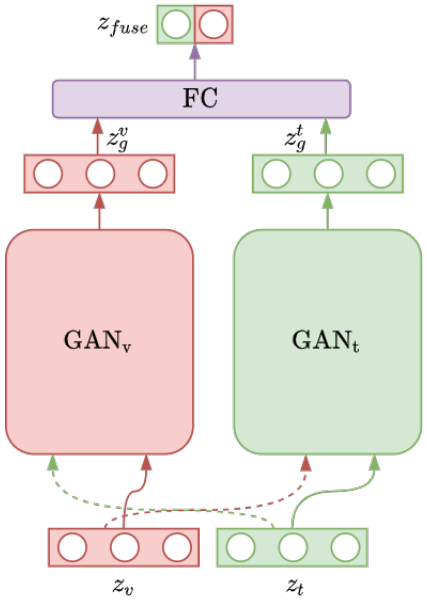


Figure 2: The overall GAN-Fusion architecture. FC denotes the feed-forward layer, which accepts the individual generator outputs and outputs a final fused vector z_{fuse}.

We first clean p_t using the Ekphrasis tool [2], which applies advanced text tokenization techniques to process hashtags and emoticons, and also corrects common typographical errors. For instance, refer to the following example:

"@fiery_eyes, this is soooo coool borther! ;) #coolforever" ⟶ "[user] fiery_eyes [/user] this is so cool brother! [wink] [hashtag] cool forever [/hashtag]."

Notice that the tokenizer adds appropriate tags for users, hastags and emoticons; corrects spelling mistakes ("borther" ⟶ "brother"); segments concatenated words ("coolforever" ⟶ "cool forever"); and handles elongated texts ("coool" to "cool").

We also perform part-of-speech tagging (POS-tagging) on the clean text and construct a (subject, object, verb, modifier) tuple for a given sentence. Additionally, we use Fast R-CNN [10], a light-weight object detection module, to identify different acting entities involved in p_v. Extracting entities from both p_t and p_v allows the model to gain a contextual understanding of p as well, and as the model is trained on more examples, it learns to measure the degree of association between a given entity composition (from both image and text) and the presence of hate.

3.3 Fusion modules

Since multimodal data is highly heterogeneous, we use two adaptive fusion mechanisms to effectively model inter- and intra-modal dynamics [40, 41]. In addition to addressing heterogeneity, these architectures perform impressively on the task of multimodal fusion, despite having significantly fewer number of parameters than the transformers.

GAN-Fusion. GAN-Fusion [40] employs two architecturally similar adversarial modules–one for image and one for text–to fuse latent vectors from different modalities. Figure 1 (b) shows the architecture of GAN_t, the GAN-Fusion module for text. It has two main components: a generator G and a discriminator D. The generator G takes $\boldsymbol{z_t}$ as input along with some random normal noise, and outputs $\boldsymbol{z_g}$, the generated latent vector. We assign $\boldsymbol{z_g}$ a *false* label and $\boldsymbol{z_v}$ a *true* label. The discriminator D takes a vector – either $\boldsymbol{z_v}$ or $\boldsymbol{z_g}$ – as input. We denote D's input as $\boldsymbol{z_d}$.

During training, the task of the generator is to match its output latent code $\boldsymbol{z_g}$ as closely as possible to $\boldsymbol{z_v}$. On the other hand, the discriminator tries to determine if $\boldsymbol{z_d} = \boldsymbol{z_g}$ (in which case, it outputs *false*) or $\boldsymbol{z_d} = \boldsymbol{z_v}$ (in which case, it outputs *true*). Note that D has no way to know if its input is $\boldsymbol{z_g}/\boldsymbol{z_v}$ beforehand as it only sees a fixed-size vector as input. To summarize, the generator tries to fool the discriminator, while the discriminator tries to tell apart the difference between its inputs $\boldsymbol{z_g}$ and $\boldsymbol{z_v}$. Therefore, the overall adversarial objective of GAN_t is given as follows:

$$\min_G \max_D J^t_{adv}(D,G) = \mathbb{E}_{x\sim p_{\boldsymbol{z_v}}(x)}[\log D(x)] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z_t}}(\boldsymbol{z})}[\log(1 - D(\boldsymbol{z_g}))] \tag{1}$$

Equation 1 only describes the objective of GAN_t; however, GAN-Fusion module for the visual modality, GAN_v, has the same architecture as GAN_t. It only differs in the input parameters. Its generator takes $\boldsymbol{z_v}$ as input, and tries to match it with $\boldsymbol{z_t}$. Therefore, we assign a *true* label to $\boldsymbol{z_t}$ and as before, a *false* label to $\boldsymbol{z_g}$. The task of the discriminator remains the same: distinguish between the two different types of its inputs ($\boldsymbol{z_t}$ and $\boldsymbol{z_g}$ in this case). Hence, the objective function for GAN_v can also be expressed in a similar fashion to GAN_t so the overall objective function of the GAN-Fusion module is $J_{adv} = J^t_{adv} + J^v_{adv}$.

Both GAN_t and GAN_v output separate latent codes denoting their respective learned distributions. Hence, GAN-Fusion uses a feed-forward or a fully-connected layer to combine generated latent vectors of the modality-specific adversarial modules. It takes the different latent codes as input and outputs a latent vector, $\boldsymbol{z_{fuse}}$, which is finally used for the downstream task of hate speech detection. Figure 2 shows the complete architecture of the GAN-Fusion module.

Auto-Fusion. Auto-Fusion [40] uses an autoencoder-type architecture to promote information retention during the fusion step. It aims to maximize the correlation between the fused and the input latent vectors. Figure 1 (a) shows the architecture of Auto-Fusion module. It consists of an encoder, which takes the individual modalities' latent codes as input and tries to reconstruct the input using a lower-dimensional latent vector. The intermediate lower-dimensional latent vector is used as $\boldsymbol{z_{fuse}}$ for the downstream task of hate speech detection. The reconstruction loss for Auto-Fusion is given as follows:

$$J_{auto} = ||\ \hat{z} - z\ ||^2 \tag{2}$$

where $z = [\boldsymbol{z_v}; \boldsymbol{z_t}]$ is the concatenation of $\boldsymbol{z_v}$ and $\boldsymbol{z_t}$. In summary, the autoencoder-type framework serves two purposes: 1) compress input to retain only the most important cues from the two modes, and 2) maximize correlation between the fused and the input latent codes. While Auto-Fusion focuses more on the inter-modal dynamics, it is still a robust architecture for hate speech detection.

3.4 Final Classification Layer

After obtaining $\boldsymbol{z_{fuse}}$ from the fusion module, we pass it through a final classification layer. It is a feed-forward layer, followed by a softmax activation function. The cross-entropy loss is given by:

$$J_C = -\sum_{l\in labels} t(l)\log y(l) \tag{3}$$

where $t(\cdot)$ is the ground-truth distribution, $y(\cdot)$ is the predicted probability from the softmax layer.

Figure 3 shows the end-to-end pipeline of the working model. First, an input publication from an online social platform, along with its captioned text, passes through the feature extraction step. Separate extractors for the image and the text modality are used. The image feature extractor first detects the participating entities involved in the image using object detection techniques. For instance, in the example shown in Figure 3, the image feature extractor first detects the entity 'skunk' and outputs a feature vector containing other semantic information about the image. Similarly, the text feature extractor identifies prominent entities in the text such as the participating subject, object, verb, modifier and the overall sentiment of the text.

In the second step, the text and image features are fused using a fusion module. This module combines information from the two modalities to cognitively determine if the original input image exhibits online abuse. The fusion module then outputs a *Hate*/*NoHate* label for binary classification or a valid label for multi-class classification. Since we train a single end-to-end model for the task, the final objective reflects all the sub-objectives as well:

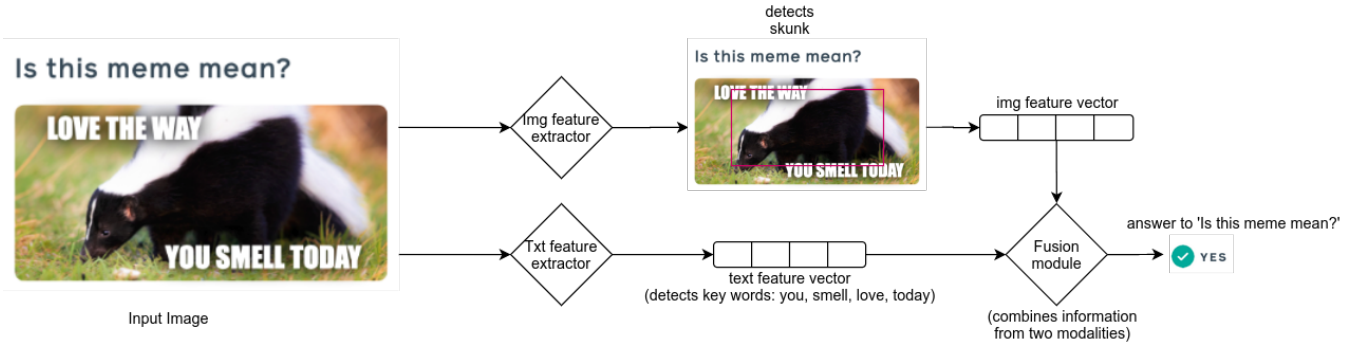


Figure 3: End-to-End pipeline of the proposed multi-agent system (example image from Kiela et al. [23])

$$J = J_C + J_F \tag{4}$$

where J_C is the classification loss described in Equation 3, and J_F is loss of the fusion module. $J_F = J_{adv}$ or $J_F = J_{auto}$ depending the type of fusion module used.

4 EXPERIMENTS AND RESULTS

In this section, we discuss different experimental setups to gauge the efficacy of our model for social good. First, we confirm the effectiveness of our model on the task of speech emotion recognition. These results serve to confirm the value of our basic architecture for combining various modalities towards classifying content accurately (and for an ultimate outcome that also has important social value). After this, we then show how our model assists with the application in focus, that of hate speech detection.

Evaluation Metrics. Before describing our classification experiments, we establish the following evaluation metrics:

- **Precision:** It is the ratio of true positives (TP) and the summation of true and false positives (TP+FP). For a class, say *Hate*, TP is the set of samples correctly classified as *Hate* and FP is the set of instances where the model incorrectly labels a sample as *Hate*.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{5}$$

- **Recall:** It is the ratio of true positives (TP) and the summation of true positives and false negatives (TP+FN). TP is the same as explained earlier, and FN is the set of instances where the model incorrectly predicts the label to be other than *Hate*.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{6}$$

- **F1-Score:** It is the harmonic mean of Precision and Recall. While Precision indicates the portion of correct predictions, Recall tells us about the total portion of instances actually retrieved. F1-score is an important metric as it gauges the *overall* performance of the system.

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision+Recall}} \tag{7}$$

- **Accuracy:** It is defined as the fraction of true instances out of all the instances.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{8}$$

Considering the example from before, TN denotes the set of true negatives, i.e., instances where the model correctly labels a sample as not *Hate*.

For multi-class classification (>2 classes involved), we compute the above metrics for each class and then use macro-averaging to obtain a final score. We now discuss our experiments.

4.1 Speech Emotion Recognition

Speech emotion recognition involves detection of a person's emotional state based on what they say (indicated by the textual modality) and how they speak (indicated by the auditory modality). It links directly with depression detection, hence, its potential for social good is also immense. A growing cohort of healthcare professionals use such emotion recognition techniques as an aiding tool to treat patients suffering from clinical depression and PTSD [14, 38]. Hence, we perform a comprehensive set of experiments on this task[2].

Model	Input modes	Fusion type	P	R	F	A
E1	audio	none	57.3	57.3	57.3	56.6
E2	text	none	71.4	63.2	67.1	64.9
BiL1	text	none	53.2	40.6	43.4	43.6
BiL2	audio+text	Concat	66.1	65.0	65.5	64.2
E3	audio+text	Concat	72.9	71.5	72.2	70.1
MDRE	audio+text	Concat	-	-	-	71.8
MHA-2	audio+text	Concat	-	-	-	76.5
M1	audio+text	Auto-Fusion	75.3	77.4	76.3	77.8
M2	audio+text	GAN-Fusion	**77.3**	**79.1**	**78.2**	**79.2**

Table 1: Precision (P), Recall (R), F1-score (F), and Accuracy (A) for emotion recognition on IEMOCAP. Note: empty cells denote that the metric was not reported in the original paper

For our experiments, we use the IEMOCAP dataset [3], which originally introduces seven emotion classes, but we only use the

[2]The classes in this case concern emotions, not hate and the channel other than text is audio (not visuals) but the basic multimodal architecture which we promote remains the same.

angry, happy, sad and *neutral* classes, appropriately merging other samples following Sahu [40]. The dataset provides access to a raw audio vector, its transcribed text, and facial expression of the speaker, but we ignore the facial expressions.

Unimodal setting. Our unimodal experiments consist of two settings: audio-only and text-only. For the audio-only setting, we extract eight hand-crafted features from the raw audio input following Sahu [39]. Then, we use an ensemble of Random Forest (RF), Gradient Boosting (XGB), and multi-layer perceptron (MLP) for classification. For the text-only setting, we perform two types of experiments. For the first set of text-only experiments, we use the TF-IDF vectors as features and use an ensemble of RF, XGB, MLP, Multinomial Naive Bayes (MNB), and Logistic Regression (LR) for classification. For the second set of text-only experiments, we convert raw text into word embeddings and pass them through a bi-directional LSTM with word-level attention (attn) model for emotion detection.

Multimodal setting. Our multimodal experiments also use two types of approaches. In the first approach, we concatenate the hand-crafted audio features and the TF-IDF vectors from the unimodal setting, and employ an ensemble of RF, XGB, MLP, MNB, and LR for emotion recognition.

For the second approach, we generate spectrogram images for the raw audio files. We then use a VGG module for encoding. For text, we compute word embeddings and use an LSTM encoder identical to our second set of text-only experiments. Next, we fuse the output signals from the two encoders, followed by a final classification layer. Note that we do not need an object detection module for the spectrograms.

Results. We compare the following architectures for speech emotion recognition:

- **MDRE** [57]: The Multimodal Dual Recurrent Encoder baseline employs RNNs to extract features from both audio and text. It uses concatenation to fuse unimodal representations.
- **MHA-2** [56]: The Multihop Attention Mechanism-2 baseline applies cross-modal attention mechanisms twice to identify the most important tokens for a given audio vector. It uses recurrent encoders to obtain latent representations of audio and text, and uses concatenation for fusion.
- **E1**: An ensemble of audio-only RF, XGB and MLP models.
- **E2**: An ensemble of text-only RF, XGB, MLP, MNB, LR.
- **E3**: An ensemble of RF, XGB, MLP, MNB, LR models for the multimodal setting, using concatenation for fusion.
- **BiL1**: The BiLSTM text-only model with attn.
- **BiL2**: A BiLSTM model, which fuses the hand-crafted audio features and TF-IDF vectors using concatenation.
- **M1**: The multimodal system using VGG to encode audio spectrograms and a BiLSTM+attn encoder for text. It uses Auto-Fusion as the fusion module.
- **M2**: It has the same architecture as M1 but uses GAN-Fusion to combine features from VGG and BiLSTM+attn.

Table 1 compiles the performance of the aforementioned models. We observe that the text-only models (E2, BiL1) are better than the audio-only model (E1). This may be due to fewer audio features as compared to text. Expectedly, all models from the multimodal setting outperform the unimodal architectures, except BiL2, which lacks an attn mechanism. We also note that M2, which uses GAN-Fusion, is the most successful system across all evaluation metrics. While M2 is the most successful system, both M1 and M2 outperform MDRE and MHA-2, which shows the superiority of our fusion mechanisms over concatenation.

4.2 Hate Speech Detection

Section 4.1 confirms the effectiveness of our basic architecture towards accurate classification in a multimodal setting. We now proceed to describe our experiments on hate speech detection.

We use the MMHS150K dataset [11], which contains 150K labelled publications, for the hate speech detection experiments. Each publication features an image, its captioned text, and a textual component; however, the captioned text may be absent from some samples. Every sample is assigned to one of the following six classes: *Racist, Sexist, Homophobic, Religion-based, No Hate,* and *Other Hate*. Despite five categories for hate, the dataset is highly skewed, with $> 80\%$ samples belonging to the *No Hate* class[3]. Hence, we include a two-class variant of the problem in our experiments, where we merge samples from the five hate classes into one *Hate* class.

Unimodal setting. For unimodal experiments, we only use the textual part of a publication. We clean the input sentences before computing word embeddings and finally passing them through a BiLSTM+attn encoder. We also perform POS-tagging on the clean text to obtain a <*subject, object, verb, modifier*> tuple. The final classification layer exploits this tuple along with the encoder's output z_t to predict a label (e.g. *Hate*, *Not Hate*) for a sample publication.

We first train the model on the relatively easier binary setting, where the model performs a *Hate*/*No Hate* classification. Keeping the rest of the architecture unchanged, we also run experiments in the full multi-class setting. To understand the importance of captioned text, we run a separate experiment, where we combine the image's caption (and not the image itself) with the textual component.

Multimodal setting. In our multimodal experiments, we encode the images using a VGG, which outputs z_v. We then use a Fast R-CNN to detect objects in the image. Simultaneously, we process the captioned and actual textual component as described in the unimodal setting. Finally, we combine the visual and textual feature vectors using an appropriate fusion module. We run experiments on the full multi-class setting, while experimenting with different types of fusion operations. Section 3 describes the functioning of various components in more detail and Figure 3 depicts the final end-to-end pipeline.

Results. We compare the following systems for hate speech detection:

- **FCM** [11]: The Feature Concatenation Model uses an Inception v3 module [53] to encode images and an LSTM layer to encode text. As the name suggests, it uses concatenation for fusion.
- **SCM** [11]: The Spacial Concatenation Model introduces a new feature map after FCM's Inception module to learn better visual representation.

[3]Label distribution: No Hate - 81.7%; Racist - 8.6%; Sexist - 2.5%; Homophobic - 2.8%; Religion-based - 0.1%; Other Hate - 4.2%

Model	Classes	Input modes	Fusion type	P	R	F
BiL	binary	text	none	70.08	63.31	66.52
TKM	binary	image+text+caption	Concat	-	-	70.1
SCM	binary	image+text+caption	Concat	-	-	70.2
FCM	binary	image+text+caption	Concat	-	-	70.4
BiL	multi	text	none	45.18	33.4	38.41
BiL	multi	text+caption	none	45.38	33.67	38.67
VBiL	multi	image+text+caption	Concat	55.27	35.54	43.04
VBiL	multi	image+text+caption	Auto-Fusion	59.65	43.87	50.56
VBiL	multi	image+text+caption	GAN-Fusion	**61.33**	**51.34**	**55.89**

Table 2: Precision (P), Recall (R), F1-score (F) for multimodal hate speech detection on MMHS150K. Note: empty cells denote that the metric was not reported in the original paper

- **TKM** [11]: The Textual Kernels Model trains multiple kernels in addition to the feature maps in SCM to capture multimodal interactions more expressively.
- **BiL**: The unimodal BiLSTM+attn model for encoding text. Notably, it uses the Ekphrasis tool for text-cleaning.
- **VBiL**: The multimodal system employs VGG to encode images and a BiLSTM+attn encoder for text. It also uses the Ekphrasis tool to clean the textual component.

We use Precision, Recall, and F1-Score to evaluate performance of different architectures described above. Table 2 compiles the results of our hate speech experiments [4]. It includes information about input modalities involved, the experiment setting (binary/multi) and the type of fusion module used. These results are preliminary, and form part of our work in progress on this solution. For binary classification experiments, we first observe that unimodal BiL is highly competitive with the multimodal FCM, SCM, and TKM baselines. Since LSTM is common to all of them, we attribute the impressive performance to text cleaning and the powerful attention mechanism. The multi-class experiments show an expected drop in performance. This is also partly due to a high class imbalance in the dataset. Interestingly, image captions have no significant effect by themselves on the performance of the model (as shown by BiL's multi-class experiments). However, inclusion of images (rows where 'Inputs modes' is image+text+caption) boosts classification performance. This shows that our effort to reason about both text and image together in order to detect hate speech has been effective [5].

5 CONCLUSION, DISCUSSION AND FUTURE WORK

As presented in Figure 3, the two primary modalities of text and image processing need to coordinate their activities to effectively detect possible hate speech in online networks. The architecture that we have presented here offers a novel direction for integrating these two agents for classifying content as hate. This multiagent system will serve to inform users invested in adjusting the presentation of content online (be it to flag what might be questionable or to remove it from the stream) and as such this work is therefore focused on the overall aim of improving social good (mitigating harm inflicted by users who are not pro-social).

Figure 4 shows our envisaged design of the process to moderate content. The platform admin is the end user, with all required privileges to control the inflow and outflow of content on the platform. She can add new content to the platform or edit/delete existing material at her discretion. Once our system detects potentially hateful content, the platform admin can be notified.

We view our connection to agent-based processing to be aligned with those of other researchers who seek to yield improvements in the social conditions of individuals, once more effective processing of the online environments of these users has been achieved. This approach has been adopted, for instance, in work published at AAMAS regarding delivering better influence maximization in uncertain social networks in order to assist homeless youths (e.g. Kamarthi et al. [21], Yadav et al. [54]) where the algorithms designed, once run, can help to propose recommendations for actions in the real-world which will achieve social good.

To summarize, in this paper, we sketch a solution to the problem of hate speech detection through a multi-agent system. Section 4 demonstrates the effectiveness of the architecture proposed for coordinating text processing and image processing and offers an improved analysis of online content. We show our model works for a key classification task (that also has social value): effectively detecting emotion. The central value of using a combination of modalities in this application area is the kind of promise that we seek to build upon in order to deliver new advances for the very important social problem of online hate speech detection. The results of Section 4.2 present the current analysis of our algorithms within this particular context. We see avenues for future work from this particular work in progress, in order to refine the processing towards greater performance.

Improved GANs. Despite the high-quality generation, GANs suffer from the "mode-collapse" problem, wherein they fail to capture entire modes in the real data distribution. For instance, a GAN trained on the MNIST dataset – a collection of handwritten digits from one to ten – might neglect a subset of digits from its output. Furthermore, training GANs can be tricky sometimes [1, 45]. An interesting line of solutions includes an implicit maximum likelihood estimation training objective [29–31], which insists on the use of full-recall GANs.

[4]Due to the dataset being more highly skewed, accuracy is not listed.
[5]These results are preliminary. Next steps include running VBiL in a binary setting and using more powerful encoders for text and images

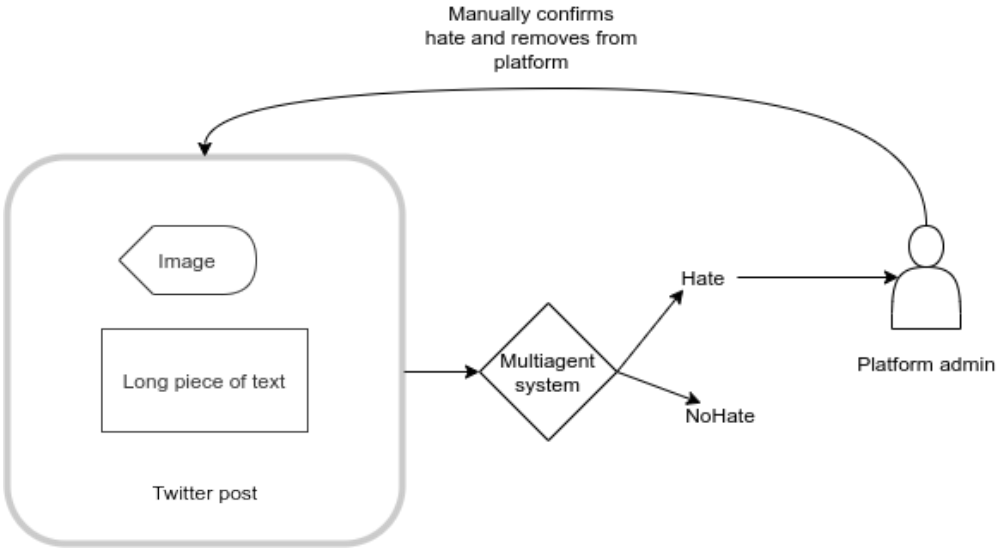


Figure 4: How to use the proposed multiagent system to moderate content online.

Embedding External Knowledge of the World. Modelling context is crucial for multimodal hate speech detection. In many cases, neither the text nor the image are hateful individually, but acquire such meaning in combination. These challenging instances require the system to possess an external knowledge of the world. Therefore, dynamic knowledge sources such as Google Vision's Web Entity Detection API[6], which can tackle the ever growing pool of information on the web, can immensely enhance the performance of the system. Another line of approaches includes effectively exploiting the stored knowledge in the neural network's parameters [28].

REFERENCES

[1] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. 2017. Generalization and equilibrium in generative adversarial nets (gans). In *ICML*. PMLR, 224–232.

[2] Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, 747–754.

[3] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. IEMOCAP: Interactive emotional dyadic motion capture database. *Language resources and evaluation* (2008), 335.

[4] Jie Chen, Gang Liu, and Xin Chen. 2019. AnimeGAN: A Novel Lightweight GAN for Photo Animation. In *International Symposium on Intelligence Computation and Applications*. Springer, 242–256.

[5] Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. 2012. Detecting offensive language in social media to protect adolescent online safety. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*. IEEE, 71–80.

[6] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[8] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 1999. Learning to forget: Continual prediction with LSTM. (1999).

[6]https://cloud.google.com/vision/docs/detecting-web

[9] Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2015. Multilingual language processing from bytes. *arXiv preprint arXiv:1512.00103* (2015).

[10] R Girshick. 2015. Fast r-cnn. arXiv 2015. *arXiv preprint arXiv:1504.08083* (2015).

[11] Raul Gomez, Jaume Gibert, Lluis Gomez, and Dimosthenis Karatzas. 2019. Exploring Hate Speech Detection in Multimodal Publications. *arXiv preprint arXiv:1910.03814* (2019).

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[13] Edel Greevy and Alan F Smeaton. 2004. Classifying racist texts using a support vector machine. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 468–469.

[14] Lang He and Cui Cao. 2018. Automated depression analysis using convolutional neural networks from speech. *Journal of biomedical informatics* 83 (2018), 103–111.

[15] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. In *Advances in neural information processing systems*. 4565–4573.

[16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[17] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360* (2016).

[18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1125–1134.

[19] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled Representation Learning for Non-Parallel Text Style Transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 424–434. https://doi.org/10.18653/v1/P19-1041

[20] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410* (2016).

[21] Harshavardhan Kamarthi, Priyesh Vijayan, Bryan Wilder, Balaraman Ravindran, and Milind Tambe. 2019. Influence maximization in unknown social networks: Learning Policies for Effective Graph Sampling. *arXiv preprint arXiv:1907.11625* (2019).

[22] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4401–4410.

[23] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *arXiv preprint arXiv:2005.04790* (2020).

[24] Oren Z Kraus, Ben T Grys, Jimmy Ba, Yolanda Chong, Brendan J Frey, Charles Boone, and Brenda J Andrews. 2017. Automated analysis of high-content microscopy data with deep learning. *Molecular systems biology* 13, 4 (2017), 924.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[26] Dhruv Kumar, Robin Cohen, and Lukasz Golab. 2019. Online abuse detection: the value of preprocessing and neural attention models. In *Proceedings of the Tenth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. 16–24.

[27] Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting tweets against blacks. In *Twenty-seventh AAAI conference on artificial intelligence*.

[28] Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401* (2020).

[29] Ke Li and Jitendra Malik. 2018. Implicit Maximum Likelihood Estimation. *ArXiv* abs/1809.09087 (2018).

[30] Ke Li and Jitendra Malik. 2018. On the Implicit Assumptions of GANs. *ArXiv* abs/1811.12402 (2018).

[31] Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. 2018. PacGAN: The Power of Two Samples in Generative Adversarial Networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) *(NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 1505–1514.

[32] Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminarasimhan, Paul Pu Liang, AmirAli Bagher Zadeh, and Louis-Philippe Morency. 2018. Efficient Low-rank Multimodal Fusion With Modality-Specific Factors. In *ACL*. 2247–2256. https://doi.org/10.18653/v1/P18-1209

[33] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*. 1412–1421. https://doi.org/10.18653/v1/D15-1166

[34] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[35] Niklas Muennighoff. 2020. Vilio: State-of-the-art Visio-Linguistic Models applied to Hateful Memes. *arXiv preprint arXiv:2012.07788* (2020).

[36] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. 2011. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 689–696.

[37] Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*. 145–153.

[38] Emna Rejaibi, Daoud Kadoch, Kamil Bentounes, Romain Alfred, Mohamed Daoudi, Abdenour Hadid, and Alice Othmani. 2019. Clinical Depression and Affect Recognition with EmoAudioNet. *CoRR* (2019).

[39] Gaurav Sahu. 2019. Multimodal Speech Emotion Recognition and Ambiguity Resolution. *arXiv preprint arXiv:1904.06022* (2019).

[40] Gaurav Sahu. 2020. *Adaptive Fusion Techniques for Effective Multimodal Deep Learning*. Master's thesis. University of Waterloo.

[41] Gaurav Sahu and Olga Vechtomova. 2021. Adaptive Fusion Techniques for Multimodal Data. *To appear in EACL* (2021).

[42] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Advances in neural information processing systems*. 2234–2242.

[43] Anna Schmidt and Michael Wiegand. 2017. A Survey on Hate Speech Detection using Natural Language Processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*. Association for Computational Linguistics, Valencia, Spain, 1–10. https://doi.org/10.18653/v1/W17-1101

[44] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[45] Mathieu Sinn and Ambrish Rawat. 2018. Non-parametric estimation of jensen-shannon divergence in generative adversarial network training. In *AISTATS*. 642–651.

[46] Nitish Srivastava and Ruslan R Salakhutdinov. 2012. Multimodal learning with deep boltzmann machines. In *NeurIPS*. 2222–2230.

[47] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[48] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.

[49] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. Multimodal transformer for unaligned multimodal language sequences. In *ACL*, Vol. 2019. 6558.

[50] Riza Velioglu and Jewgeni Rose. 2020. Detecting Hate Speech in Memes Using Multimodal Deep Learning Approaches: Prize-winning solution to Hateful Memes Challenge. *arXiv preprint arXiv:2012.12975* (2020).

[51] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Generating videos with scene dynamics. In *Advances in neural information processing systems*. 613–621.

[52] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. 2018. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 0–0.

[53] Xiaoling Xia, Cui Xu, and Bing Nan. 2017. Inception-v3 for flower classification. In *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*. IEEE, 783–787.

[54] Amulya Yadav, Hau Chan, Albert Xin Jiang, Haifeng Xu, Eric Rice, and Milind Tambe. 2016. Using Social Networks to Aid Homeless Shelters: Dynamic Influence Maximization under Uncertainty.. In *AAMAS*, Vol. 16. 740–748.

[55] Shuai Yang, Zhangyang Wang, Zhaowen Wang, Ning Xu, Jiaying Liu, and Zongming Guo. 2019. Controllable Artistic Text Style Transfer via Shape-Matching GAN. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[56] Seunghyun Yoon, Seokhyun Byun, Subhadeep Dey, and Kyomin Jung. 2019. Speech Emotion Recognition Using Multi-hop Attention Mechanism. In *ICASSP*. 2822–2826.

[57] Seunghyun Yoon, Seokhyun Byun, and Kyomin Jung. 2018. Multimodal speech emotion recognition using audio and text. In *SLT Workshop*. IEEE, 112–118.

[58] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2017. Tensor Fusion Network for Multimodal Sentiment Analysis. In *EMNLP*. 1103–1114. https://doi.org/10.18653/v1/D17-1115

[59] Ron Zhu. 2020. Enhance Multimodal Transformer With External Label And In-Domain Pretrain: Hateful Meme Challenge Winning Solution. *arXiv preprint arXiv:2012.08290* (2020).