

Twitter Image and Text Data Analysis for identifying Derogatory Posts

Ahel Das Chatterjee, Devtanu Misra, Anish De, Sounak Majumder, Malay Gain, Sayak Rana

Abstract

This paper argues that in order to capture the full spectrum of emotions expressed on Twitter, a multimodal analysis approach is required that incorporates both text and image analysis techniques, as they together can provide a more comprehensive understanding of the emotions and sentiments expressed by users on the platform. To address the challenge of capturing recent sentiment in multiple languages, we created a novel dataset called Sentweet, which consists of a collection of Twitter posts along with their associated images and texts in multiple languages. To gather the data for the Sentweet dataset, we utilized Twitter APIs for web scraping and also employed optical character recognition (OCR) and image captioning techniques to extract text from the images while also preserving the original images. This approach enabled us to collect a diverse set of Twitter posts with both textual and visual components, which are crucial for conducting multimodal sentiment analysis. Additionally, we also applied several preprocessing techniques to ensure the quality and consistency of the data before conducting sentiment analysis. For the model architecture of Sentweet, we utilized Transformers, which have emerged as a state-of-the-art deep learning approach for natural language processing tasks due to their ability to capture long-term dependencies and contextual information in text data. We compared the performance of our Sentweet model with other popular deep learning approaches such as LSTM and CNN and found that Transformers outperformed these models in terms of accuracy and efficiency.

Keywords: keyword one, keyword two

PACS: 0000, 1111

2000 MSC: 0000, 1111

1. Introduction

Social media platforms like Twitter have become a prominent source of information and communication in today’s world. With millions of active users sharing their thoughts, opinions, and emotions on a daily basis, these platforms have emerged as a valuable source of data for researchers and businesses seeking to understand public sentiment and opinion.

Among various social media platforms, Twitter has become a popular choice for sentiment analysis due to its massive user base and real-time nature. Twitter sentiment analysis involves analyzing the sentiments and emotions expressed by users on the platform towards a particular topic, product, or service. This analysis can provide valuable insights into public opinion and help businesses and individuals make informed decisions.

However, conducting sentiment analysis on Twitter poses several challenges due to the platform’s diverse linguistic and visual cues. Traditional approaches like natural language processing (NLP) may not be sufficient to capture the full spectrum of emotions expressed on Twitter, as users often include images or emojis in their posts to convey their emotions. To overcome this challenge, multimodal sentiment analysis techniques that incorporate both textual and visual data have emerged as a promising solution.

In this paper, we present a comprehensive study of Twitter sentiment analysis using a multimodal approach. Specifically, we introduce a novel dataset called Sentweet, which consists of Twitter posts along with their associated images and texts in multiple languages. The posts are downloaded from twitter using their developer APIs and then preprocessed for better training of machine learning model. We also propose a deep learning-based sentiment analysis model for Sentweet using Transformers, which outperforms traditional models like LSTM and CNN. Our study also points out the comparison of different machine learning base models and encoders used during the experiment.

Overall, our study demonstrates the importance of multimodal sentiment analysis for capturing the full spectrum of emotions expressed on Twitter and provides a valuable resource for researchers and practitioners in the field of sentiment analysis.

Additionally, our study contributes to the growing body of literature on multimodal sentiment analysis and highlights the need to develop techniques that can effectively handle cross-lingual sentiment analysis on Twitter. Moreover, our approach can be extended to other social media platforms like

Instagram, which poses unique challenges for sentiment analysis due to its visual-centric nature.

Through our study, we aim to provide a better understanding of public sentiment on social media platforms and help businesses and individuals make informed decisions based on the insights gained from sentiment analysis.

2. Related Work

The use of social media data for sentiment analysis has been extensively studied in the literature. Several studies have focused on sentiment analysis using only text data and have achieved promising results using various machine learning and deep learning techniques, such as Support Vector Machines (SVMs), Random Forest, Recurrent Neural Networks (RNNs), and Convolutional Neural Networks (CNNs) (Read et al. [1], 2005; Go et al. [2], 2009; Davidov et al. [3]; Zhang et al. [4]; Moraes et al. [5], 2013; Zhang et al. [6], 2018; Tang et al. [7], 2019). However, the inclusion of visual data has been shown to improve sentiment analysis accuracy, particularly for images and videos (You et al. [8]; Linger et al. [9], 2018; Li et al. [10], 2019).

Recently, there has been a growing interest in multimodal sentiment analysis, which aims to capture sentiment and emotion using both text and visual data. Several studies have proposed multimodal approaches for sentiment analysis, including models that combine features extracted from both text and images (Poria et al. [11], 2017; Huang et al. [12], 2019; Huang et al. [13], 2020; Ortis et al. [14], 2021), as well as models that use deep learning techniques to jointly model both modalities (Hu et al. [15]; Mahendhiran et al. [16]; Hazarika et al. [17], 2018; Li et al. [10], 2019).

In the context of Twitter sentiment analysis, there have been several studies that have proposed approaches to handle the diverse linguistic and visual cues present on the platform. For example, (Kumar et al. [18], 2018; Zhao et al. [19], 2019) proposed a multimodal approach that combines visual features from images with textual features from tweets using SVMs for sentiment analysis. Similarly, (Zhang et al. [20], 2022) proposed a CNN-based model that combines text and image features for Twitter sentiment analysis.

However, to the best of our knowledge, there has been limited work on creating a dataset that includes both textual and visual components of Twitter posts in multiple languages. In this paper, we address this gap by introducing the Sentweet dataset and propose a deep learning-based sentiment analysis model using Transformers that outperforms traditional models for sentiment

analysis on Twitter. Our work builds upon the existing literature by demonstrating the effectiveness of a multimodal approach for sentiment analysis on Twitter, highlighting the importance of including visual data, and providing a valuable resource for researchers and practitioners in this field.

3. Dataset

A brand new dataset Sentweetv1 was prepared for this research work. It has around 3500 rows.

The Sentweetv1 dataset is collected from twitter. Twitter accounts of news pages, controversial personalities were scrapped using the Twitter Developer API [21]. For every tweet scrapped, the caption of the post, the media url of the post, language of the text and the post id were collected. The media were downloaded from the url using curl and stored in a folder. The name of the media file was marked as the Image ID of a row. 3

To improve the quality of dataset and the model, the image OCR and image caption were extracted from the media files. The post id column was filled with the id of the original post. Twitter has content moderation policies in place, so the percentage of negative posts were really less. To balance the dataset, a synthetically generated dataset was used. Vidgen et. al. [22] prepared a synthetic dataset of about 40000 entries. A sample of around 1000 entries were extracted from this dataset and appended to Sentweetv1.

Then five new columns were added: Hate Speech, Profanity, Sentiment, Insult and Derogatory. The first four were manually annotated to prepare a dataset. The Derogatory column was annotated in the following way: if any two of Hate Speech, Profanity, Sentiment or Insult was true/negative, it was labelled as derogatory, otherwise not. 3

3.1. Image OCR

Pytesseract tool was used to capture the OCR from the media files. Python tesseract is a wrapper class for Google’s Tesseract-OCR Engine.

Table 1: Data scraped from Twitter

Column Name	Description
POST_ID	The id of the twitter post
POST_LINK	URL of the scraped tweet
Text	The caption of the post
MEDIA_URL	The URL where the associated media is stored
Images_ID	The file name of the downloaded media (unique for each row)
Language	English and Hindi posts were scraped

Table 2: Manually annotated columns in Sentweet Dataset

Column Name	Labels
Hate Speech	TRUE or FALSE
Profane	TRUE or FALSE
Sentiment	Positive, Negative or Neutral
Targeted Insult	TRUE or FALSE
Derogatory	TRUE or FALSE
Text Type	'Caption', 'Image OCR' or 'Image Caption'

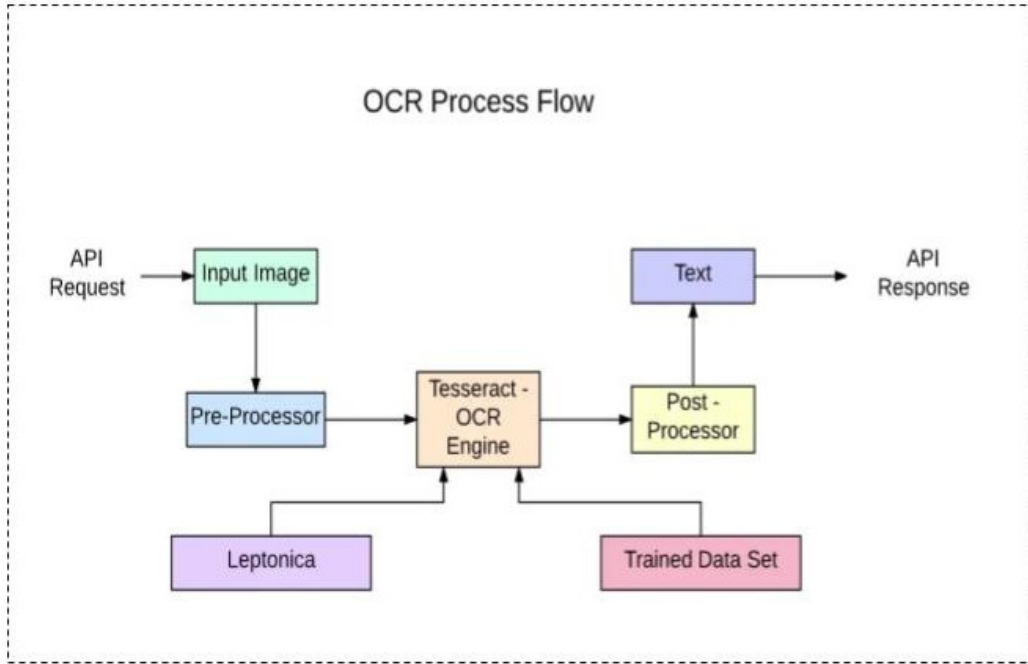
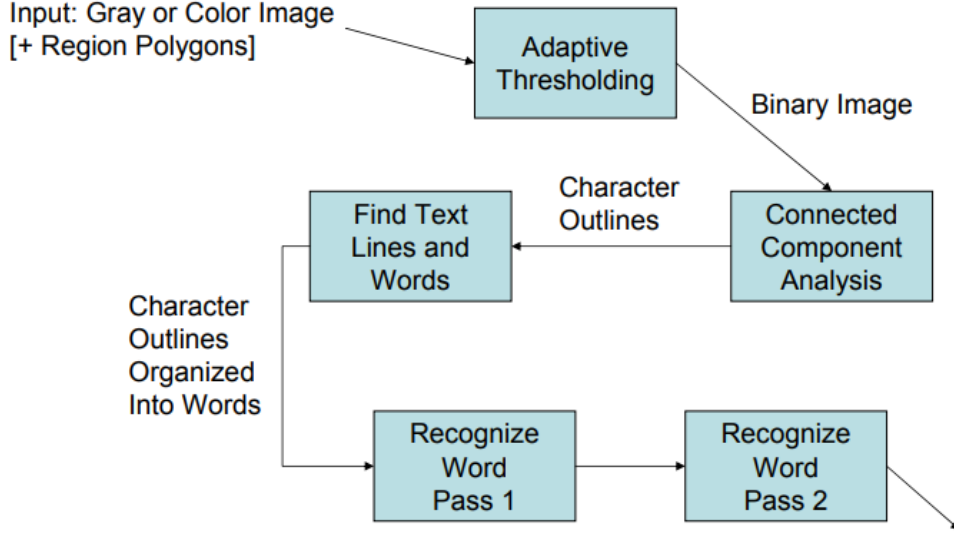


Figure 1: A rough sketch of OCR capture using PyTesseract

Text blobs are categorized into lines, which are then examined for fixed pitch or proportional text. The text lines are segmented into words based on the type of character spacing utilized. For fixed pitch text, the words are chopped instantly by character cells. In contrast, for proportional text, words are segmented using spaces and fuzzy spaces [23]. The image recognition is achieved in two passes. In first pass, the OCR engine attempts to recognize each word. In the second pass, the words that were not recognized properly are recognized again.



[24]

Figure 2: Tesseract OCR Engine architecture

3.2. Image Caption

Image captioning is the task of generating a textual description of an image. It has important applications in fields such as image retrieval, video surveillance, and assistive technologies for the visually impaired. In recent years, deep learning models have achieved remarkable success in image captioning, with the Flickr dataset being a popular benchmark for evaluating such models.

3.2.1. Flickr Dataset

The Flickr dataset consists of over 8,000 images, each with five human-generated captions. The images cover a wide range of topics and are diverse in terms of their content and composition. The dataset has been widely used for training and evaluating image captioning models, and its popularity has led to the development of several benchmarking frameworks.

3.2.2. Image Caption model

The Transformers package is an open-source library for natural language processing (NLP) developed by Hugging Face. It provides state-of-the-art

implementations of popular NLP models, such as BERT, GPT-2, and T5, and also includes pre-trained models for several NLP tasks. The package is built on top of PyTorch, a popular deep learning framework, and provides a user-friendly interface for training and evaluating NLP models.

The image-to-text pipeline in the Transformers package is a combines two models: an image encoder and a language decoder. The image encoder is a convolutional neural network (CNN) that takes an image as input and produces a feature vector that captures the visual content of the image. The language decoder is a transformer-based model that takes the feature vector as input and generates a natural language text description of the image.

The image encoder is typically pre-trained on a large dataset of images, such as Flickr, using a variant of the SGD algorithm, such as Adam or RMSprop. The language decoder is typically pre-trained on a large corpus of text, such as Wikipedia or Common Crawl, using the transformer architecture and a variant of the masked language modeling objective.

During fine-tuning, the image encoder and language decoder are combined and trained end-to-end on a dataset of image-caption pairs. The model is trained to minimize the cross-entropy loss between the predicted caption and the ground-truth caption. The model is typically fine-tuned using a variant of the Adam algorithm [25], with a learning rate schedule and early stopping to prevent overfitting. Adam algorithm uses the following to update the weights:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (1)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (2)$$

$$m_t^{corr} = \frac{m_t}{1 - \beta_1^t} \quad (3)$$

$$v_t^{corr} = \frac{v_t}{1 - \beta_2^t} \quad (4)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_t^{corr}} + \epsilon} m_t^{corr} \quad (5)$$

t is the current iteration step

θ represents the network weights

g_t is the gradient of the objective function with respect to the weights at iteration t

m_t and v_t are estimates of the first and second moments of the gradient, respectively

β_1 and β_2 are decay rates for the first and second moments, respectively (typically set to 0.9 and 0.999)

ϵ is a small value added for numerical stability (typically set to 10^{-8})

η is the learning rate.

3.3. Data Preprocessing

The synthetic data present in Sentweetv1 had no media files. So most relevant and negative images obtained from a google search of a particular hate speech entry was downloaded and saved in the folder.

All entries without any media files were dropped. Further noises, like emojis, non-English texts, etc were removed as well.

Corresponding to a particular post id, all the texts in the "Text" column were combined in one single row, i.e, the post caption, image OCR and image captions were all combined to give a single row. All columns except the following were dropped: "**Text**", "**IMAGES_ID**" and "**Derogatory**". This dataset is called Sentweetv2 and was used for further analysis. The texts were then tokenized.

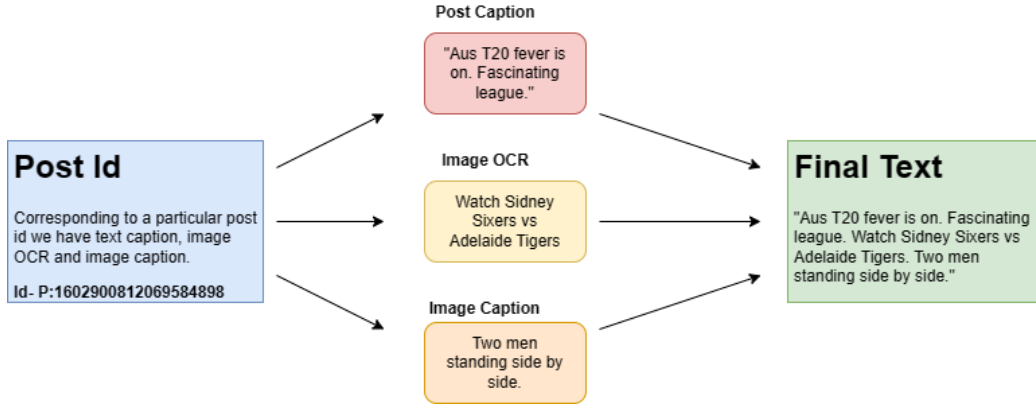
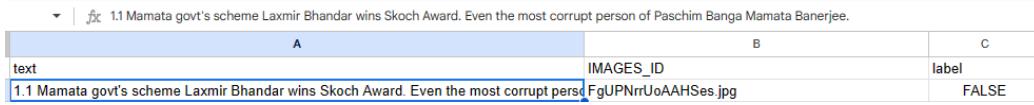


Figure 3: Merging the rows with same Post Id into one

Tokenization is the process of breaking a stream of text into words, phrases, symbols, or other meaningful elements called tokens [26]. **Stop words** were removed from these tokens. Stop words are words that barely affect the nature of a sentence, like "a", "an", "the", etc which are ignored by search engines during a search.



A	B	C
text	IMAGES_ID	label
1.1 Mamata govt's scheme Laxmir Bhandar wins Skoch Award. Even the most corrupt person of Paschim Banga Mamata Banerjee.	FgUPNrrUoAAHSes.jpg	FALSE

Figure 4: Screenshot of a dataset entry

4. Methodology

4.1. Architecture for this research work

We are proposing the following multimodal architecture where we are taking a tweet as an input. The tweets contained both image and text. The tweet was then broken into several components and passed through different layers of the architecture to finally predict whether the tweet is derogatory or not. The default and prevalent definition of derogatory post was assumed. If some tweets contained profane, targeted insult, hate speech or negative emotion towards living or non living beings, it was labeled as derogatory.

Two inputs were fed into the model:

1. The textual input which was the concatenation of the texts from tweet caption, image caption and image OCR.
2. The image of the tweet, if present.

This pipeline implemented a multimodal classification model that accepted in text and image inputs and gave a class label as the output. The architecture consists of several components:

Text encoder using BERT: The input text was first encoded using the BERT model. The BERT model was initialized with pre-trained weights from the 'bert-base-uncased' model. The BertTokenizer was also initialized to preprocess the input text into tokenized sequences. An encoded vector of size 768 was obtained as an output.

Image encoder using Vision Transformer: The input image was encoded using the Vision Transformer model. The '*vit_base_patch16_224*' pre-trained model from the timm library was used to encode the image. This also gave a feature vector of size 768 as output.

Cross-modal attention: The cross-modal attention mechanism was used to combine the output from the text and image encoders. This mechanism involves computing attention scores between the two inputs and using the attention scores to weight the image input. The weighted image was then combined with the text input using a residual connection. The encoded text was used as a query and the encoded image was used as a key. The combined output was a feature vector of size 768.

Transformer encoder: The output from the cross modal attention layer was then passed through a transformer layer which had six pytorch transformer layers to learn the features.

Fully connected layer for classification: The output from the Transformer encoder was then passed through a fully connected layer with a ReLU activation function. The output from this layer was then passed through another fully connected layer with a Softmax activation function to obtain the class probabilities.

Dropout: The model includes a dropout layer with a dropout rate of 0.2 to prevent overfitting.

The forward() method was used to run the inputs through the model. The text input was first tokenized using the BertTokenizer and encoded using the BERT model. The image input was encoded using the Vision Transformer model. The attended tensor was then obtained by applying cross-modal attention to the output from the text and image encoders. The attended tensor was then passed through the Transformer encoder and the fully connected layer to obtain the class probabilities.

Finally, the model was trained using a loss function CrossEntropyLoss and an optimizer. The model was trained using backpropagation and the gradients were updated using the optimizer. Gradient clipping was used to prevent

exploding gradient and improving the stability of the system.

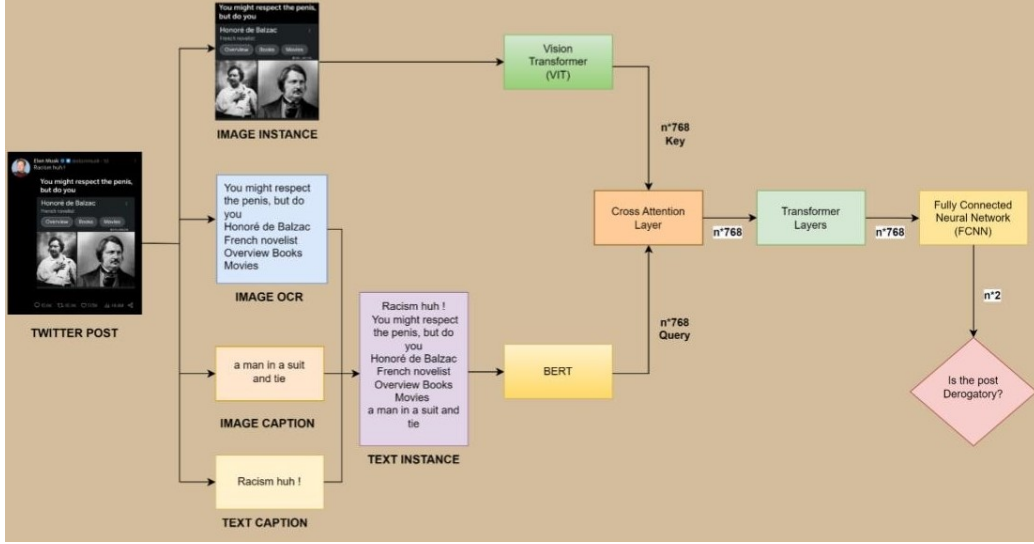


Figure 5: Proposed Architecture For Our Research Work

4.2. BERT

BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based neural network architecture used for natural language processing tasks, including text encoding. BERT takes in a sequence of tokenized input text and applies a sequence of self-attention and feed-forward layers to generate a sequence of fixed-length representations, or embeddings[27].

The self-attention layer in BERT can be expressed mathematically as:

$$Q = XW_q \quad K = XW_k \quad V = XW_v \quad A = \text{softmax}(QK^T) \quad H = AV \quad (6)$$

where X is the input sequence of token embeddings, W_q , W_k , and W_v are learnable weight matrices, Q , K , and V are the query, key, and value matrices, respectively, and A is the attention scores.

The feed-forward layer in BERT can be expressed mathematically as:

$$H = \text{LN}(M_1 H_{\text{prev}} + M_2) \quad A = \text{gelu}(H) \quad H_{\text{new}} = \text{LN}(M_3 A + M_4) \quad (7)$$

where M_1 , M_2 , M_3 , and M_4 are learnable weight matrices and biases, `gelu` denotes the Gaussian error linear unit activation function, LN denotes layer normalization, and H_{prev} is the previous layer's output.

The final output of the BERT encoder is obtained by applying a pooling operation over the sequence of embeddings. BERT uses two types of pooling operations: [CLS] token pooling and mean pooling. In [CLS] token pooling, the embedding of the first token, which is typically the special token [CLS], is used as the output feature vector. In mean pooling, the embeddings of all tokens are averaged to obtain a single feature vector. Both pooling operations result in an output feature vector of size 768.

In summary, the BERT encoder can be expressed as:

$$H_{\text{final}} = \text{Pool}(H) \quad y = H_{\text{final}} W_{\text{out}} \quad (8)$$

where Pool denotes either [CLS] token pooling or mean pooling, H is the sequence of embeddings, W_{out} is a learnable weight matrix, and y is the output feature vector of size 768.

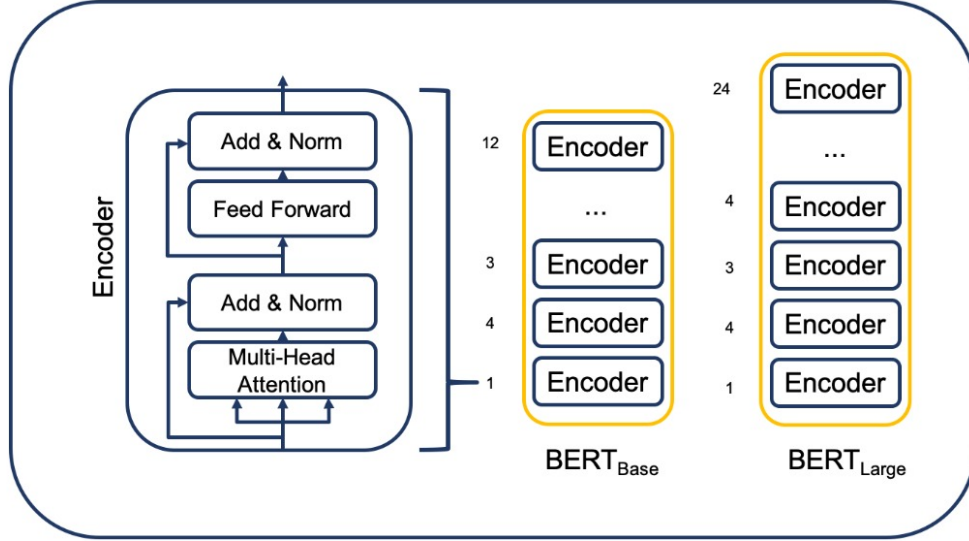


Figure 6: BERT Encoder Architecture

4.3. VIT

VIT (Vision Transformer) is a transformer-based neural network architecture used for image encoding [28]. VIT applies a sequence of self-attention

and feed-forward layers to a sequence of flattened image patches to generate a sequence of fixed-length representations, or embeddings [29]. At first the input image is divided into non-overlapping patches, and each patch is flattened into a vector of length p [30]. The input patch vectors are then linearly transformed into queries (Q), keys (K), and values (V) using learnable weight matrices W_q , W_k , and W_v , respectively. This can be expressed mathematically as:

$$Q = XW_q \quad K = XW_k \quad V = XW_v \quad (9)$$

where X is the sequence of flattened image patches, and W_q , W_k , and W_v are learnable weight matrices of size $(p \times d)$. The queries, keys, and values are then fed into the multi-head self-attention module, which computes the attention scores (A) and the output embeddings (H) as follows:

$$A = \text{softmax} \left(\frac{QK^T}{\sqrt{d_h}} \right) \quad H = AV \quad (10)$$

where d_h is the dimension of the queries, keys, and values after being split into n_{heads} number of heads. A is the attention scores obtained by applying softmax to the scaled dot-product between the queries and keys. H is the weighted sum of the values using the attention scores as weights. The output embeddings from the self-attention layer are passed through a two-layer feed-forward neural network with a GELU activation function and layer normalization. This can be expressed mathematically as:

$$H' = LN(M_2(GELU(M_1(H)))) \quad (11)$$

where M_1 and M_2 are learnable weight matrices and biases, LN denotes layer normalization, $GELU$ denotes the Gaussian error linear unit activation function, and H is the output embeddings from the self-attention layer. Positional encoding is added to the output embeddings to inject information about the spatial arrangement of the patches in the image. This can be expressed mathematically as:

$$H'' = H' + PE \quad (12)$$

where PE is a learnable positional encoding matrix of the same size as H' . The above steps are repeated for a fixed number of transformer layers to generate a sequence of embeddings of the same length as the input

patch sequence. The sequence of embeddings is then fed into a mean pooling operation to obtain a fixed-length representation of the image. This can be expressed mathematically as:

$$H_{final} = \frac{1}{N} \sum_{i=1}^N H_i \quad y = H_{final} W_{out} \quad (13)$$

where N is the number of embeddings in the sequence, H_i is the i -th embedding, H_{final} is the mean of all embeddings, and W_{out} is a learnable weight matrix. The final output feature vector y had a size of 768.

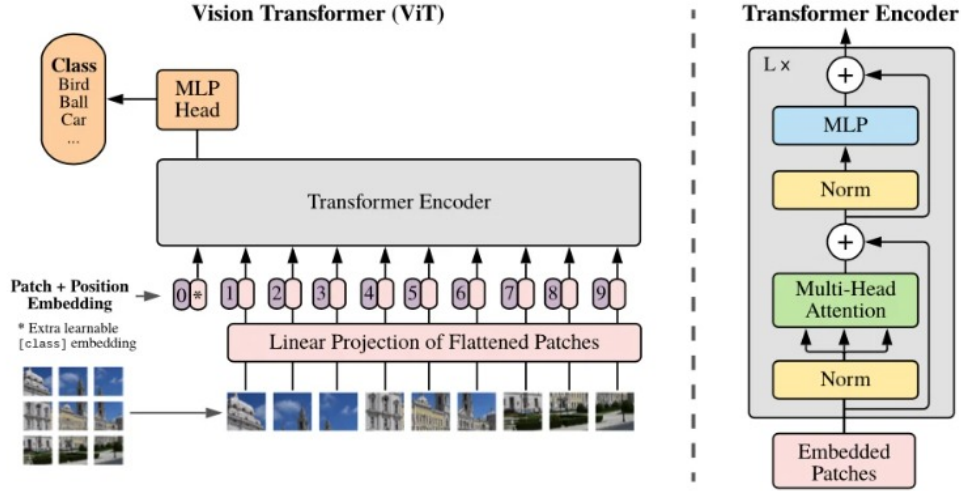


Figure 7: ViT Encoder Architecture

4.4. Cross Attention Layer

In the context of natural language processing and computer vision, cross-attention is a technique used in transformer-based models to combine information from two different modalities, such as text and image [31]. In this mechanism, the attention scores are calculated between the query tensor from one modality and the key tensor from another modality. The output tensor is then computed as a weighted sum of the value tensor from the second modality, where the weights are determined by the attention scores.

In this model, two input tensors, key tensor x_1 and query tensor x_2 , were

generated from Image Encoder (ViT) and Text Encoder (BERT) which are of size 768 each. The cross-attention was achieved using the following steps:

Query and Key Generation: The input x_1 was passed through an attention query layer to generate a query tensor q . Similarly, the input x_2 was passed through an attention key layer to generate a key tensor k .

Attention Scores: The attention scores were computed by taking the dot product between the query tensor and the transpose of the key tensor. This was achieved by the following formula:

$$attn_{scores} = q \cdot k^{\top} \quad (14)$$

Attention Weights: The attention scores were normalized using the softmax function to get the attention weights. The attention weights were then used to compute a weighted sum of the value tensor x_2 to generate the attended tensor $x_{2_{attended}}$. This was achieved by the following formulas:

$$attn_{weights} = softmax(attn_{scores}) \quad (15)$$

$$x_{2_{attended}} = attn_{weights} \cdot x_2 \quad (16)$$

Residual Connection and Linear Layer: The attended tensor $x_{2_{attended}}$ was added to the input tensor x_1 using a residual connection. The output of the residual connection was then passed through a linear layer to generate the final output tensor out . This was achieved by the following formulas:

$$combined = x_1 + x_{2_{attended}} \quad (17)$$

$$out = linear(combined) \quad (18)$$

Overall, the cross-attention mechanism allowed the model to attend to relevant parts of x_2 given the information in x_1 . The resulting output tensor out had a size of 768, which was the same size as the input tensors.

4.5. Pytorch Transformer Layers

The PyTorch Transformer is a neural network architecture used for natural language processing tasks such as machine translation and language modeling [32]. It consists of several transformer layers, with each layer composed of two sub-layers: a multi-head self-attention mechanism and a position-wise fully connected feed-forward [?]. Here’s a brief explanation of each layer along with their formulas:

Multi-Head Self-Attention Mechanism: This sub-layer took a sequence of input vectors X of size 768 and applied an attention mechanism to it, resulting in an output sequence of the same length. This was achieved through the following steps:

Query, Key, and Value Generation: The input sequence X was first passed through three different linear layers to generate query, key, and value tensors Q , K , and V , respectively. This was done using the following formulae:

$$Q = XW_Q \quad K = XW_K \quad V = XW_V \quad (19)$$

where W_Q , W_K , and W_V are learnable weight matrices of size 768×768 .

Scaled Dot-Product Attention: The query tensor Q was used to attend to the key tensor K and generate an attention weight tensor A . This was done using the following formulae:

$$A = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) \quad (20)$$

where d_k is the dimensionality of the key vectors. In this case, $d_k = 768$.

Weighted Sum of Values: The attention weight tensor A was then used to generate a weighted sum of the value tensor V , resulting in an output tensor O . This was done using the following formula:

$$O = AV \quad (21)$$

The resulting tensor O has the same size as the input tensor X , i.e., 768.

Residual Connection and Layer Normalization: The output tensor

O was added to the input tensor X using a residual connection, and the resulting tensor was normalized using layer normalization.

Position-Wise Fully Connected Feed-Forward Network: This sub-layer applied a fully connected feed-forward network to each position of the input sequence independently. This was done using the following formulae:

$$F_1 = \text{relu}(XW_1 + b_1) \quad F_2 = F_1W_2 + b_2 \quad (22)$$

where W_1 , W_2 , b_1 , and b_2 are learnable weight matrices and biases of size 768×3072 and 3072 , respectively, and relu is the rectified linear unit activation function.

Residual Connection and Layer Normalization: The output of the feed-forward network was added to the input tensor using a residual connection, and the resulting tensor was normalized using layer normalization.

The output of each transformer layer was then passed to the next layer until the final layer, which produced the output of the transformer. For a six-layer transformer, the input tensor of size 768 went through six transformer layers, and the output tensor of the last layer was also of size 768.

5. Experimental Analysis

We used a combination of BERT and ViT models for the detection of derogatory posts on Twitter. BERT is a pre-trained model that is widely used for natural language processing tasks. ViT, on the other hand, is a pre-trained model that is used for image classification tasks. However, recent studies have shown that ViT can also be used for natural language processing tasks.

Several combinations of text and image encoder models were fine-tuned using our customized Twitter data set, Sentweetv2. The fine-tuning process involved training the models on the training set and evaluating their performance on the testing set. The Sentweetv2 dataset had around 800 entries, of which about 600 were used for training and the rest were used for testing.

It was observed that BERT 8 and DistilBERT 9 gave exceptionally good

results when combined with VIT as the image encoder. Both of them gave an accuracy of 87% on the testing dataset and a F1 Score of 0.87. These two text transformer models gave identical results.

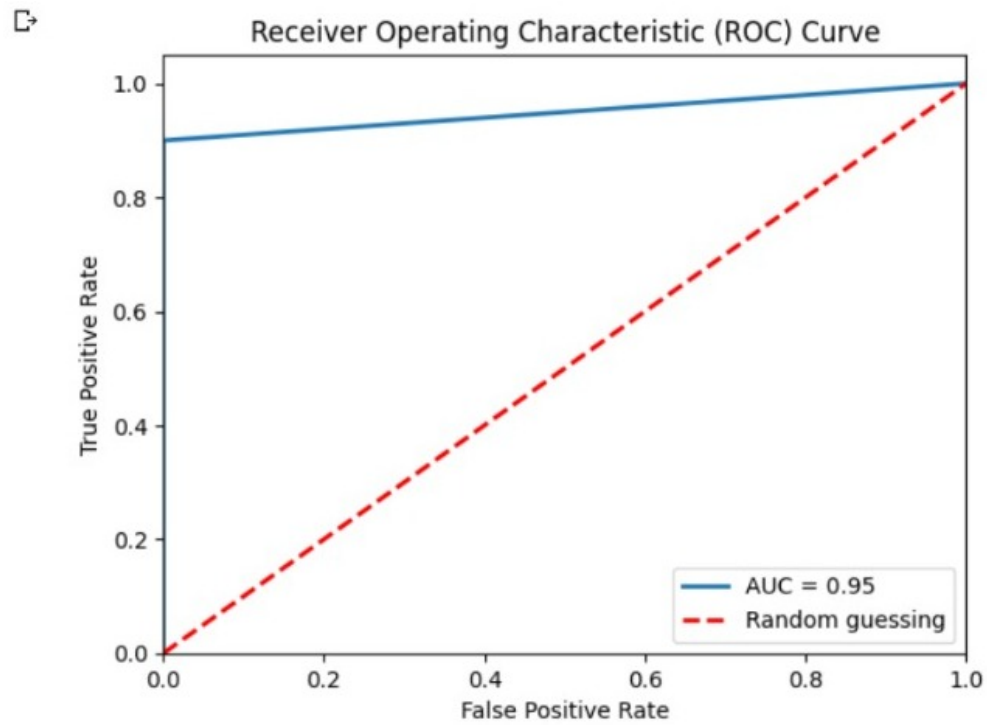


Figure 8: AUC ROC Curve for BERT+VIT

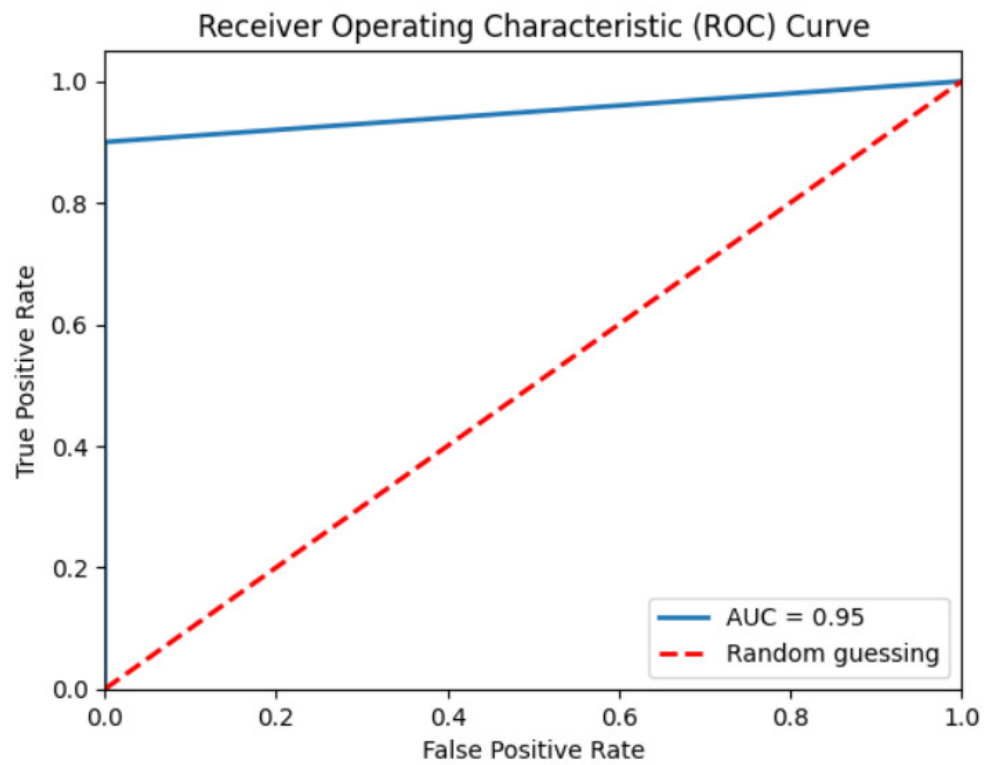


Figure 9: AUC ROC Curve for DistilBERT+ViT

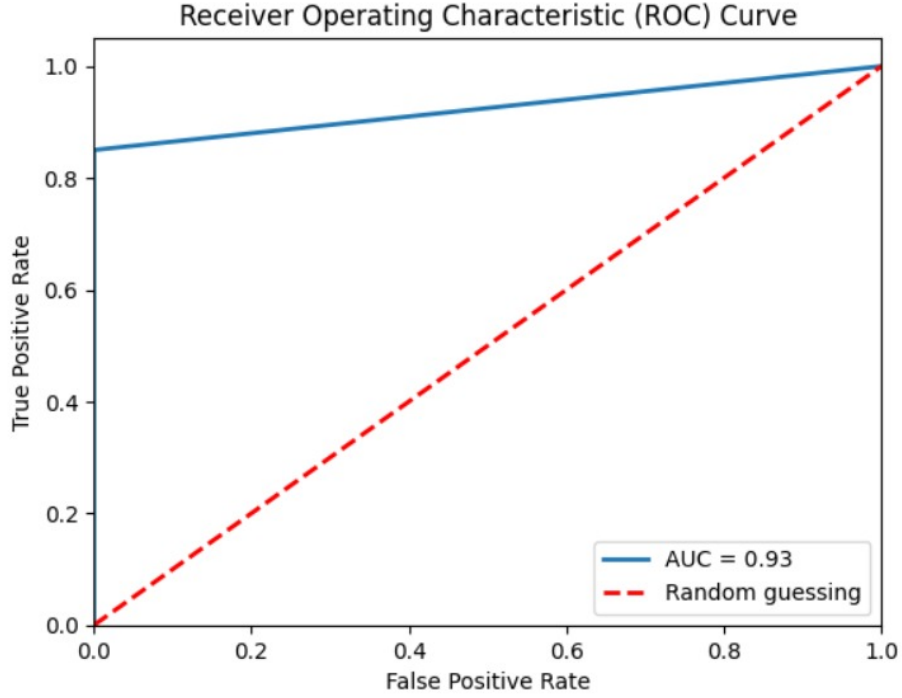


Figure 10: AUC ROC Curve for ALBERT+VIT

But the results deteriorated drastically when the image encoder was changed from VIT to BEIT. Accuracy reduced to 72% and F1 score was 0.53. If we look at its AUC-ROC curve 11, it is observed that the AUC curve merged with the random guessing plot. Thus, VIT was the image encoder model that gave the best results for the combinations we used for our analysis.

Table 3: Comparison of Models on the Basis of Results

Text Encoder	Image Encoder	Accuracy	Precision	Recall	F1 Score
BERT	VIT	87%	0.88	0.85	0.87
ALBERT	VIT	85%	0.87	0.82	0.84
ALBERT	BEIT	72%	0.56	0.50	0.53
DistilBERT	VIT	87%	0.88	0.85	0.87

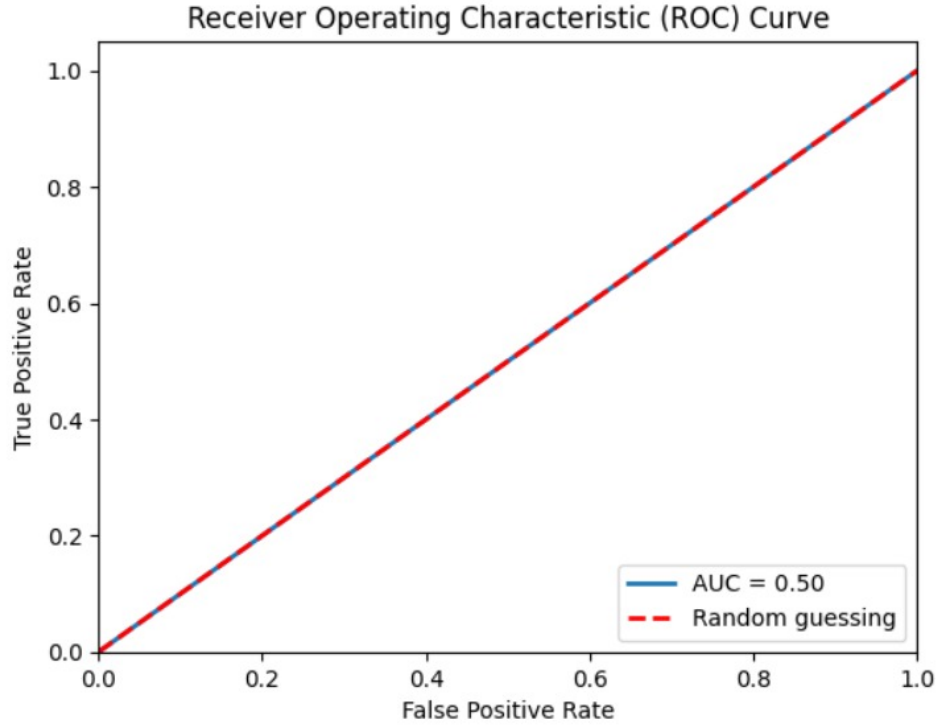


Figure 11: AUC ROC Curve for ALBERT+BEIT

A pipeline was created and the predictions for different texts were noted as shown in 12. The texts marked in red were predicted as derogatory by the pipeline created using our proposed architecture. In this image, the 0

signified non-derogatory post while 1 signified a derogatory post. It predicted "cartoon of a person on a computer screen" as derogatory, which is not really true. But in all other cases, it predicted correctly. There were some insults aimed at races, gender identity of individuals and religions, which were correctly flagged as derogatory.

```

https://t.co/2ZlspvHfag cartoon of a person on a computer screen FoSa#1laEAAAC6E.jpg
tag 89/1 https://t.co/2ZlspvHfag man with a hat on F0m1ny5ak000N-.jpg
Last Time#INDVAUS #BorderGavaskarTrophy https://t.co/sM2jmx089a collage of photos showing a man in a suit and a woman in a dress F0lwnWWeakAA1NIC.jpg
Nagpur Pitch https://t.co/v1FAVYb6cEa man is playing a game of baseball F0kz2bbaAE0L3o.jpg
It makes no sense to point fingers at an industrialist if he is not breaking any law. If anyone is found guilty th44,-A; https://t.co/Du7XrRVK9f M4vhaR1qYqUgI1
RT : Hockey Special Train during FIH Hockey Men's World Cup -2023 https://t.co/1nCHdyu18v F0Um_hFaYAEtyJP.jpg
Glimpses from Jagaddhatri Puja Inauguration at Dumdum. I pray to Maa Jagaddhatri Raj Rajeshwari for the well-being44,-A; https://t.co/B1E4iy70kr FgFaxOnVIAAHCF
M4rhe hhi iaana hai... https://t.co/5KfHtWaa5a man holding a sign with a cartoon character F0fzDe1a0AAdCm1ng
because they deserve at least a taste of it for helping the white liberals and Jews push it on us I do n't think we should allow it to destroy them though ahe
Ravi Ashwin#INDVAUS #BorderGavaskarTrophy https://t.co/X022LRC006a man in a baseball uniform is throwing a baseball F0g1P1laIAEKwz.jpg
Kafi jor ki kahaani... https://t.co/VW00XoVlc4a man and woman are talking on a cell phone F0VZV05o041vxo2.jpg
Try El Paso Texas 500 much mudd invasion that the river is the same color as those playing/crossing in it .. year round ! sounak_images 28.jpg
a man with a beard is talking on a cell phone https://t.co/trghK4u50K F0YAE1zWIAEVH02.jpg
1.1 People of Paschim Banga now know that the TMC leaders who once used to make big claims are big thieves. We tol44,-A; https://t.co/VFgcMeYY7w Fmbu7BnaUAuG21
Eventually we wo n't be able to tell who is who. this is all sad . anish_Eventually.png
Watching those videos makes me feel like im going to explode with hatred for those creatures . sounak_images 22.jpg
Encoded
tensor([1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1], device='cuda:0')

```

Figure 12: Predictions from Pipeline

6. Conclusion

In this paper, we presented a comprehensive study of Twitter sentiment analysis using a multimodal approach. We introduced a novel dataset called Sentweet, consisting of Twitter posts along with their associated images and texts in multiple languages. We also proposed a deep learning-based sentiment analysis model using Transformers, which outperformed traditional models like LSTM and CNN. Our study demonstrated the importance of multimodal sentiment analysis for capturing the full spectrum of emotions expressed on Twitter and provided a valuable resource for researchers and practitioners in the field of sentiment analysis.

The Sentweet dataset and our proposed model have several potential applications, including monitoring brand reputation, tracking public opinion on social and political issues, and analyzing customer feedback. Future work can extend our approach by incorporating additional modalities like audio and video data or by exploring different deep learning architectures for multimodal sentiment analysis. Additionally, there is a need to address the challenges of cross-lingual sentiment analysis on Twitter and develop techniques that can handle code-switching and language variations effectively.

Moreover, while our study focuses on Twitter, our approach can be applied to other social media platforms as well. For instance, Instagram is a visual-centric platform that poses unique challenges for sentiment analysis,

and our multimodal approach can be used to address these challenges. Overall, our study contributes to the growing literature on multimodal sentiment analysis and demonstrates its importance for understanding public sentiment on social media platforms.

References

- [1] J. Read, Using emoticons to reduce dependency in machine learning techniques for sentiment classification, in: Proceedings of the ACL student research workshop, 2005, pp. 43–48.
- [2] A. Go, R. Bhayani, L. Huang, Twitter sentiment classification using distant supervision, CS224N project report, Stanford 1 (12) (2009) 2009.
- [3] D. Davidov, O. Tsur, A. Rappoport, Enhanced sentiment learning using twitter hashtags and smileys, in: Coling 2010: Posters, 2010, pp. 241–249.
- [4] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu, B. Liu, Combining lexicon-based and learning-based methods for twitter sentiment analysis, HP Laboratories, Technical Report HPL-2011 89 (2011) 1–8.
- [5] R. Moraes, J. F. Valiati, W. P. G. Neto, Document-level sentiment classification: An empirical comparison between svm and ann, Expert Systems with Applications 40 (2) (2013) 621–633.
- [6] L. Zhang, S. Wang, B. Liu, Deep learning for sentiment analysis: A survey, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8 (4) (2018) e1253.
- [7] F. Tang, L. Fu, B. Yao, W. Xu, Aspect based fine-grained sentiment analysis for online reviews, Information Sciences 488 (2019) 190–204.
- [8] Q. You, L. Cao, H. Jin, J. Luo, Robust visual-textual sentiment analysis: When attention meets tree-structured recursive neural networks, in: Proceedings of the 24th ACM international conference on Multimedia, 2016, pp. 1008–1017.
- [9] E. Kim, R. Klinger, A survey on sentiment and emotion analysis for computational literary studies, arXiv preprint arXiv:1808.03137 (2018).

- [10] X. Li, L. Bing, W. Zhang, W. Lam, Exploiting bert for end-to-end aspect-based sentiment analysis, arXiv preprint arXiv:1910.00883 (2019).
- [11] A. Zadeh, M. Chen, S. Poria, E. Cambria, L.-P. Morency, Tensor fusion network for multimodal sentiment analysis, arXiv preprint arXiv:1707.07250 (2017).
- [12] F. Huang, X. Zhang, Z. Zhao, J. Xu, Z. Li, Image-text sentiment analysis via deep multimodal attentive fusion, *Knowledge-Based Systems* 167 (2019) 26–37.
- [13] F. Huang, K. Wei, J. Weng, Z. Li, Attention-based modality-gated networks for image-text sentiment analysis, *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 16 (3) (2020) 1–19.
- [14] A. Ortis, G. M. Farinella, G. Torrisi, S. Battiato, Exploiting objective text description of images for visual sentiment analysis, *Multimedia Tools and Applications* 80 (2021) 22323–22346.
- [15] A. Hu, S. Flaxman, Multimodal sentiment analysis to explore the structure of emotions, in: *proceedings of the 24th ACM SIGKDD international conference on Knowledge Discovery & Data Mining*, 2018, pp. 350–358.
- [16] P. Mahendhiran, S. Kannimuthu, Deep learning techniques for polarity classification in multimodal sentiment analysis, *International Journal of Information Technology & Decision Making* 17 (03) (2018) 883–910.
- [17] N. Majumder, D. Hazarika, A. Gelbukh, E. Cambria, S. Poria, Multimodal sentiment analysis using hierarchical fusion with context modeling, *Knowledge-based systems* 161 (2018) 124–133.
- [18] A. Kumar, D. Kawahara, S. Kurohashi, Knowledge-enriched two-layered attention network for sentiment analysis, arXiv preprint arXiv:1805.07819 (2018).
- [19] Z. Zhao, H. Zhu, Z. Xue, Z. Liu, J. Tian, M. C. H. Chua, M. Liu, An image-text consistency driven multimodal sentiment analysis approach

- for social media, *Information Processing & Management* 56 (6) (2019) 102097.
- [20] F. Zhang, X.-C. Li, C. P. Lim, Q. Hua, C.-R. Dong, J.-H. Zhai, Deep emotional arousal network for multimodal sentiment analysis and emotion recognition, *Information Fusion* 88 (2022) 296–304.
 - [21] B. Batrinca, P. C. Treleaven, Social media analytics: a survey of techniques, tools and platforms, *Ai & Society* 30 (2015) 89–116.
 - [22] B. Vidgen, T. Thrush, Z. Waseem, D. Kiela, Learning from the worst: Dynamically generated datasets to improve online hate detection, *arXiv preprint arXiv:2012.15761* (2020).
 - [23] R. Smith, An overview of the tesseract ocr engine, in: *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, Vol. 2, 2007, pp. 629–633. doi:10.1109/ICDAR.2007.4376991.
 - [24] R. Smith, et al., Tesseract ocr engine, Lecture. Google Code. Google Inc (2007).
 - [25] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* (2014).
 - [26] K. Raj, S. Polamuri, S. S. D. V. Preethi, S. Penmetsa, S. Pilli, Prediction on sarcasm sentiment detection of twitter data, Tech. rep., EasyChair (2020).
 - [27] S. Sukhbaatar, E. Grave, G. Lample, H. Jegou, A. Joulin, Augmenting self-attention with persistent memory, *arXiv preprint arXiv:1907.01470* (2019).
 - [28] P. Zhang, X. Dai, J. Yang, B. Xiao, L. Yuan, L. Zhang, J. Gao, Multi-scale vision longformer: A new vision transformer for high-resolution image encoding, in: *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 2998–3008.
 - [29] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, D. Tao, A survey on vision transformer, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45 (1) (2023) 87–110. doi:10.1109/TPAMI.2022.3152247.

- [30] S. H. Lee, S. Lee, B. C. Song, Vision transformer for small-size datasets, arXiv preprint arXiv:2112.13492 (2021).
- [31] A. Nagrani, S. Yang, A. Arnab, A. Jansen, C. Schmid, C. Sun, Attention bottlenecks for multimodal fusion, Advances in Neural Information Processing Systems 34 (2021) 14200–14213.
- [32] D. Rothman, A. Gulli, Transformers for Natural Language Processing: Build, train, and fine-tune deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, and GPT-3, Packt Publishing Ltd, 2022.